NONLINEAR ESTIMATION FOR VISION-BASED AIR-TO-AIR TRACKING

A Thesis
Presented to
The Academic Faculty

by

Seung-Min Oh

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2007

NONLINEAR ESTIMATION FOR VISION-BASED AIR-TO-AIR TRACKING

Approved by:

Professor Eric N. Johnson,
Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Professor Anthony J. Calise
School of Aerospace Engineering
Georgia Institute of Technology

Professor Eric Feron
School of Aerospace Engineering
Georgia Institute of Technology

Professor Allen R. Tannenbaum
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor George Vachtsevanos
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 7 November 2007

To my wife, Miyoung, and our parents,

our three daughters, Soobin, Jiyoon, and Chaeyoung,

for their love, patience, and encouragement.

ACKNOWLEDGEMENTS

Above all, I would like to express my sincere gratitude to my advisor, Professor Eric N. Johnson, who first opened my eyes to nonlinear filters and advised this work, for his support, guidance, and timely advice. I learned a lot from him in all respects of education and research. This dissertation would not have been at this point without his guidance.

I would like to thank my thesis committee, Dr. Anthony J. Calise, Dr. Eric Feron, Dr. Allen R. Tannenbaum, and Dr. George Vachtsevanos for sharing their precious time and comments. I am honored to have them all in my prestigious committee members. My additional special thanks go to Dr. Calise who thoroughly read this dissertation and gave very detailed comments until the last moment even though he couldn't attend my defense presentation.

I would like to offer my thanks to my lab mates, Jincheol Ha and Yoko Watanabe. Spending the many years in the same lab, they helped me get out of my ignorance in many areas through precious discussions. I would also like to express my appreciation to Allen Wu and Dr. Ramachandra Sattigeri for sharing their precious time and efforts to proof-reading and discussions.

I am lucky to be with wonderful members of UAV Lab, past and present, Dr. Shresh K. Kannan, Dr. Jim Neidhoefer, Nimrod Rooz, Girish Chowdhary, Claus Christmann, Jennifer Sheffield, Syed Shah, Jeong Hur, Wayne Pickell, Henrik Christophersen, Alison Proctor, Gregory Ivey, Shannon Twigg, Stewart Geyer, Michael Turbe, Sumit Mishra, Tom Apker, Mike Curry. They all helped me at some points or through invaluable memories and discussions.

I am also indebted to other control group members, Dr. Bong-Jun Yang, Dongwon

Jung, and my others for their help in various ways throughout my stay at Georgia Tech. I would also like to thank Drs. Ali Turker Kutay, Chang Chen, Yoonghyun Shin for their help during their stay at Georgia Tech as former control group members.

Finally, I would like to thank my family. I could have felt much more difficulty without their love and encouragement. I would like to thank my parents for their love, support, and encouragement throughout my life. I express my deepest love and appreciation to my special four ladies: my foremost special lady, my wife Miyoung, and my three daughters Soobin, Jiyoon, and Chaeyoung, thank you for your support, encouragement, patience, sacrifice, and love. Together with you all on this rather long journey, this has been more lively, fun, encouraging, and meaningful. You kept me always on the right track.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

DGPS Differential Global Positioning System.

DOF Degree of Freedom.

EKF Extended Kalman Filter.

EMPF Extended Marginalized Particle Filter.

GNC Guidance, Navigation, and Control.

GPS Global Positioning System.

GTEdge refers to the high thrust-to-weight ratio fixed-wing aircraft owned by Georgia Tech. UAV Lab.

GTMax refers to the Yamaha Rmax helicopter owned by Georgia Tech. UAV Lab.

IMU Inertial Measurement Unit.

INS Inertial Navigation System.

MAP Maximum A Posteriori.

MC Monte Carlo.

MCMC Markov chain Monte Carlo.

MEMS Micro-Electro-Mechanical systems.

MP Modified Polar.

NED North-East-Down.

RMS Root Mean Square.

SIR Sampling Importance Resampling.

SIS Sequential Importance Sampling.

UAV Unmanned Aerial Vehicle.

UKF Unscented Kalman Filter.

UT Unscented Transformation.

WGS World Geodetic System.

WMM World Magnetic Model.

SUMMARY

Unmanned aerial vehicles (UAV's) have been the focus of significant research interest in both military and commercial areas since they have a variety of practical applications including reconnaissance, surveillance, target acquisition, search and rescue, patrolling, real-time monitoring, and mapping, to name a few. To increase the autonomy and the capability of these UAV's and thus to reduce the workload of human operators, typical autonomous UAV's are usually equipped with both a navigation system and a tracking system. The navigation system provides high-rate ownship states (typically ownship inertial position, inertial velocity, and attitude) that are directly used in the autopilot system, and the tracking system provides low-rate target tracking states (typically target relative position and velocity with respect to the ownship). Target states in the global frame can be obtained by adding the ownship states and the target tracking states. The data estimated from this combination of the navigation system and the tracking system provide key information for the design of most UAV guidance laws, control command generation, trajectory generation, and path planning.

As a baseline system that estimates ownship states, an integrated navigation system is designed by using an extended Kalman filter (EKF) with sequential measurement updates. In order to effectively fuse various sources of aiding sensor information, the sequential measurement update algorithm is introduced in the design of the integrated navigation system with the objective of being implemented in low-cost autonomous UAV's. Since estimated state accuracy using a low-cost, MEMS-based IMU degrades with time, several absolute (low update rate but bounded error in

time) sensors, including the GPS receiver, the magnetometer, and the altimeter, can compensate for time-degrading errors. In this work, the sequential measurement update algorithm in smaller vectors and matrices is capable of providing a convenient framework for fusing the many sources of information in the design of integrated navigation systems. In this framework, several aiding sensor measurements with different size and update rates are easily fused with basic high-rate IMU processing.

In order to provide a new mechanism that estimates ownship states, a new nonlinear filtering framework, called the unscented Kalman filter (UKF) with sequential measurement updates, is developed and applied to the design of a new integrated navigation system. The UKF is known to be more accurate and convenient to use with a slightly higher computational cost. This filter provides at least second-order accuracy by approximating Gaussian distributions rather than arbitrary nonlinear functions. This is compared to the first-order accuracy of the well-known EKF based on linearization. In addition, the step of computing the often troublesome Jacobian matrices, always required in the design of an integrated navigation system using the EKF, is eliminated. Furthermore, by employing the concept of sequential measurement updates in the UKF, we can add the advantages of sequential measurement update strategy such as easy compensation of sensor latency, easy fusion of multi-sensors, and easy addition and subtraction of new sensors while maintaining those of the standard UKF such as accurate estimation and removal of Jacobian matrices. Simulation results show better performance of the UKF-based navigation system than the EKF-based system since the UKF-based system is more robust to initial accelerometer and rate gyro biases and more accurate in terms of reducing transient peaks and steady-state errors in ownship state estimation.

In order to estimate target tracking states or target kinematics, a new vision-based tracking system is designed by using a UKF in the scenario of three-dimensional air-to-air tracking. The tracking system can estimate not only the target tracking

states but also several target characteristics including target size and acceleration. By introducing the UKF, the new vision-based tracking system presents good estimation performance by overcoming the highly nonlinear characteristics of the problem with a relatively simplified formulation. Moreover, the computational step of messy Jacobian matrices involved in the target acceleration dynamics and angular measurements is removed.

A new particle filtering framework, called an extended marginalized particle filter (EMPF), is developed and applied to the design of a new vision-based tracking system. In this work, only three position components with vision measurements are solved in particle filtering part by applying Rao-Blackwellization or marginalization approach, and the other dynamics, including the target nonlinear acceleration model, with Gaussian noise are effectively handled by using the UKF. Since vision information can be better represented by probabilistic measurements and the EMPF framework can be easily extended to handle this type of measurements, better performance in estimating target tracking states will be achieved by directly incorporating non-Gaussian, probabilistic vision information as the measurement inputs to the vision-based tracking system in the EMPF framework.

CHAPTER 1

BACKGROUND

The availability of direct measurements for all states in a dynamical system is rare in most practical situations. Some states may be inaccessible internal states and thus impossible to measure, or some may be very difficult to measure or even undesirable to measure since their direct measurement may require very expensive devices. In either case, we need to reconstruct full states by means of estimation since they contain all necessary information required to describe the system under investigation, and therefore provide essential knowledge needed for the GNC algorithm design of dynamical systems. Estimation, in this context, is the process of inferring or determining the "best approximation" of the full states (and possibly constant-but-unknown parameters) by using available limited measured outputs. The measured outputs usually take the form of indirect, inaccurate, uncertain, and noisy observations.

Two important areas in aerospace systems where estimation plays a significant role are navigation and tracking. *Navigation* is the estimation of the state of the platform or ownship on which a sensor or sensors are located (e.g., position and velocity of an aircraft in a navigation frame), and *tracking* is the estimation of the state of a moving object or target based on remote measurements using one or more sensors at fixed locations or on moving platforms (e.g., position and velocity of a target measured by a radar system or a vision system in a global or local frame) [3]. Figure 1.1 shows a typical flight scenario of two UAV's that combines the navigation and tracking problems, which will be the main focus in this research.[1] In addition to these two

[1]In the case of the tracking problem, the original definition includes the estimation of both a target inertial position based on remote sensing at a fixed location and a target relative position based on sensing from a moving platform. In this research, we limit our interest in the tracking problem of estimating the relative states by onboard systems of flying vehicles in air-to-air scenarios

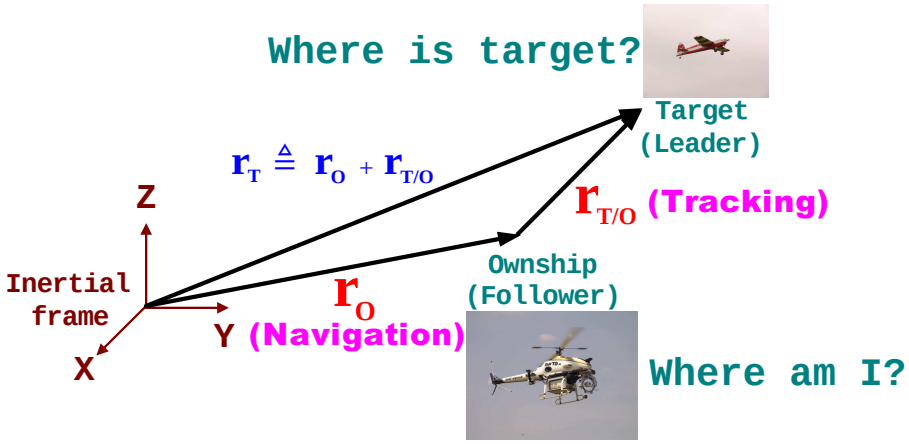
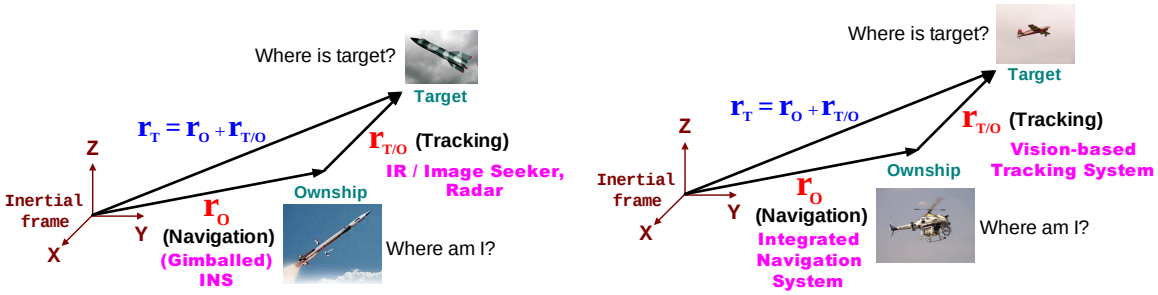


Figure 1.1: Typical Flight Scenario of Two UAV's in terms of Navigation and Tracking

problems, a number of recent research interests have focused on the determination of some parameters or characteristics of an image (e.g., feature extraction from noisy images).

The estimated states or parameters are used in various elements of aircraft GNC systems including control system design (e.g., attitude control, the design of a state feedback control law), guidance law design, and trajectory generation or path planning. Thus, estimation accuracy is a very important issue especially for the overall performance of recent sophisticated missions in modern aerospace GNC systems. The objective of this research is the development of convenient and accurate estimation frameworks for air-to-air tracking problems of UAV systems. Considering the recent sophisticated missions in modern aerospace GNC systems, the tracking problem should be solved in both local and global coordinates, which requires both ownship states (navigation) and target-ownship relative-motion states (tracking) in order to obtain the global coordinates (e.g., position and velocity of the target in a navigation frame, which can be obtained from the combination of ownship states and relative-motion states) of the target. This research focuses especially on the vision-based

since we assume no communication link to ground stations.



(a) Traditional Expensive Military Application (b) New Cost-Efficient UAV Application

Figure 1.2: Comparison of New Vision-based Tracking System with a Traditional Military Application

tracking of a flying target in air-to-air scenarios. Figure 1.2 compares this research with a traditional military application. The target characteristics are not limited, and it can be friendly or adversarial. Hence, the research here can be applied or easily extended to flying in formation with a friendly aircraft or to the target tracking of an adversarial maneuvering aircraft. In order to improve the convenience and accuracy of the estimation, the states or parameters are estimated in the framework of nonlinear filtering.

1.1 Motivation and Research Outlines

Unmanned aerial vehicles (UAV's) have drawn significant research interest in both military and commercial areas since they have a variety of practical applications including reconnaissance, surveillance, target acquisition, search and rescue, patrolling, real-time monitoring, and mapping, to name a few [9, 15, 67, 68, 122]. To increase the autonomy and the capability of these UAV's and thus to reduce the workload of human operators, a typical UAV navigation system is equipped with various sensors depending on the mission of each UAV. One indispensable capability for many autonomous UAV missions is determining the global position and velocity of target objects in the world. The knowledge of this target-state information allows autonomous UAV's to operate safely in cluttered environments with obstacles, buildings, terrain, and other

vehicles. This knowledge also becomes an essential element for the successful completion of complex mission requirements. In general, an autonomous UAV is capable of measuring its own position and velocity (ownship states) as well as those of other objects (target states) in the global coordinate system. In most UAV missions, the desired states include both the position/velocity of the autonomous ownship UAV and the position/velocity of the target objects, which can be any object of interest including stationary or moving obstacles and friendly or adversarial vehicles. The ownship states (typically inertial position, inertial velocity, and attitude) are generally provided by an integrated navigation system, which is composed of a low-cost, MEMS-based IMU and several additional sensors such as a GPS, a magnetometer, and an altimeter [92]. In addition, one of the most attractive sensor systems for measuring target tracking states (e.g., relative position and velocity with respect to the surrounding environment or other vehicles) is a vision system: typically a combination of a video camera, camera-image capturing hardware, and software for processing the captured image [42]. Figure 1.3 defines the target (global) states, (target) tracking states, and ownship states in a navigation frame used throughout this thesis. Target states in the global coordinate system can be obtained by adding the ownship states and the relative target-tracking states as shown in Figure 1.4. In other words, the estimation of target global motion can be achieved by combining ownship global motion calculated by the ownship integrated navigation system with target-ownship relative motion as processed by the vision system. The data estimated from this combination of an integrated navigation system and a vision system have become the key information for determining guidance law design, control command generation, trajectory generation, and path planning [43, 65, 98, 99, 125, 127]. In order to accurately determine the minimal set of the desired states, the problem at hand can be subdivided into two steps:

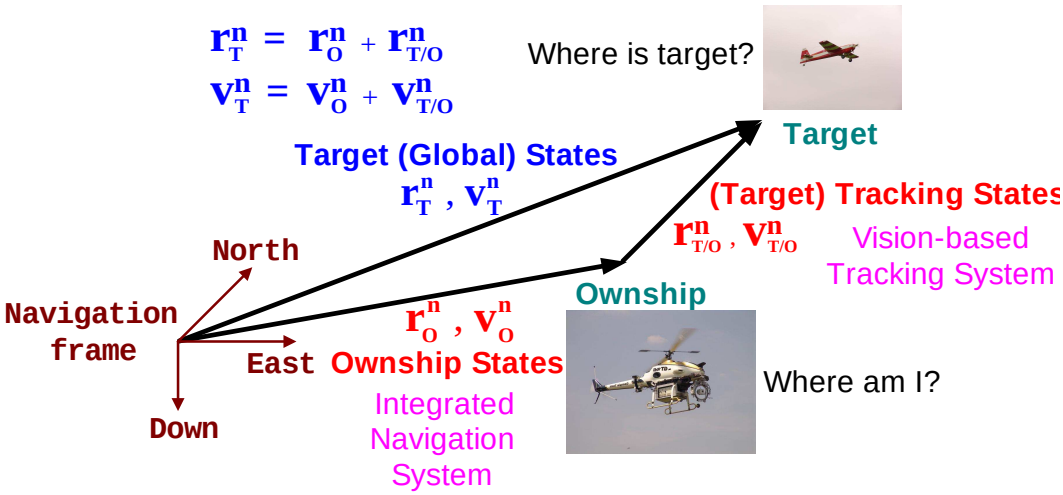


Figure 1.3: Definitions of States in a Navigation Frame

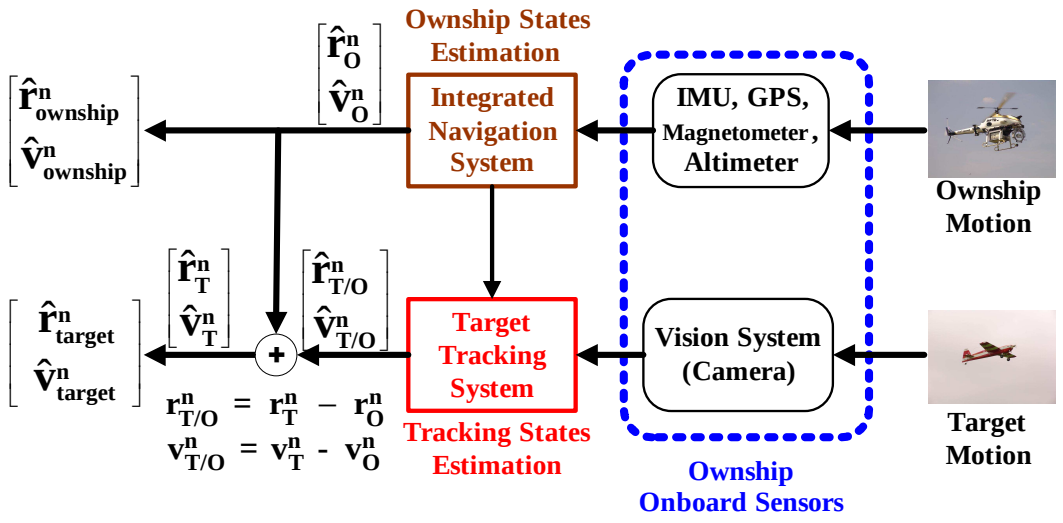


Figure 1.4: Desired States by Combining Ownship States and Target Tracking States

- how to accurately estimate the ownship states including position and velocity of the ownship UAV by using an integrated navigation system (Navigation Problem)

- how to accurately estimate the target tracking states including position and velocity of the target, given measurements of relative position of the target with respect to the ownship UAV (e.g., target azimuth and elevation angles from a vision system), and given information of ownship position and velocity provided by an integrated navigation system (this is referred to as target motion estimation or target motion analysis, Tracking Problem)

The estimation accuracy of these two problems is crucial for the safe operation of an autonomous UAV in relation to the surrounding environment and for the successful completion of sophisticated missions. For example, the accurate estimation of target states (e.g., position and velocity of stationary/moving obstacles or friendly/adversarial aircraft) allows the autonomous UAV to operate safely near obstacles or other vehicles. It also provides safe and precise mission capabilities when applied to vision-based formation flight with other aircraft. The most important building block for accurate estimation is a filter that usually performs recursive online estimation of states or parameters of interest by using a stochastic dynamical model and noisy uncertain measurements.

Since its first formulation for the linear case in 1960, the (linear) Kalman filter and its nonlinear companion, the extended Kalman filter (EKF), have become de facto standard filters due to their reliability and efficiency in many real world applications. In order to overcome the problem of the restricted application to linear Gaussian problems in the original Kalman filter, the EKF is derived from the Kalman filter based on the successive linearization, about the filter's estimated states at the previous time step, of the nonlinear process model and/or the nonlinear observation model.

Since the EKF is relatively easy to use and often provides remarkable estimations in many applications, it has become the most widely used framework to deal with nonlinear estimation problems. On the other hand, it can be surprisingly inaccurate for some practical applications depending on several system characteristics of the problem at hand including the degree of nonlinearities in the dynamical model equations (process and measurement model), the relevance in the Gaussian assumption of noise models, the amount of measurement errors, and filter initialization uncertainty. In order to overcome the limitations of the EKF, extensive research has been performed on many nonlinear filters attempting to improve estimation accuracy. One crucial issue on the development of the nonlinear filters is the computational complexity for the real-time capability of the filters. Due to the algorithm development on reducing the computational complexity and due to the hardware development on high-speed microprocessors and large capacity memory, practical nonlinear filters have become a reality in the many applications, which are traditionally known to be very difficult nonlinear estimation areas, including navigation, target tracking, and vision-based image processing.

This research focuses on the development of convenient and accurate nonlinear estimation frameworks with application to the air-to-air tracking of a friendly or adversarial target aircraft based on vision information. In order to solve this whole problem at hand, research work is broken down into the following several stages:

1. developing a nonlinear estimation framework to estimate ownship states by fusing several sources of low-cost sensor measurements (integrated navigation system)

2. developing a nonlinear estimation framework to estimate target tracking states based on deterministic vision information (deterministic vision-based target tracking system)

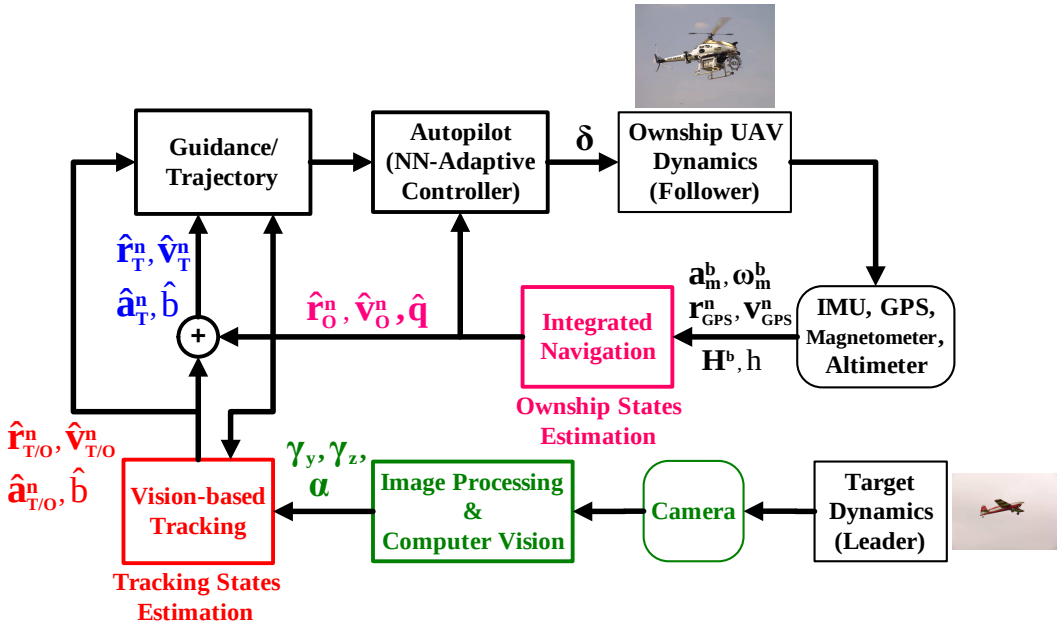


Figure 1.5: Closed-loop of Vision-Based Navigation System for Target Tracking

3. developing a nonlinear estimation framework to estimate target tracking states based on probabilistic vision information (probabilistic vision-based target tracking system)

The closed-loop diagram of the vision-based navigation for the air-to-air target tracking in this research is presented in Figure 1.5. In order to untangle the highly involved subject in this work, relevant research work is surveyed and presented in the next several sections.

1.2 Nonlinear Estimation

Estimation is the process of inferring or determining the time-varying states or constant-but-unknown parameters of dynamic systems from indirect, inaccurate, uncertain, and noisy observations. The most important ingredient of estimation in the perspective of numerical algorithms is the filter which is generally used to represent the algorithmic tool for obtaining the "best estimate" of the quantity (i.e., states or

parameters) of interest from noisy signal or data by using some algorithmic mechanisms of "filtering out" the noise. With the help of progress in several areas including probability, statistics, and computer science, one of the greatest discoveries in the history of statistical estimation theory is certainly the development of the Kalman filter. The Kalman filter is an optimal recursive estimator for estimating the states or parameters of dynamic systems in time-domain state-space formulations. Since its first exposure in the 1960's [60], innumerable research work has been performed on the subject including efforts to develop related variational/advanced theories (e.g., extended Kalman filters, unscented Kalman filters, particle filters), to implement numerically stable and efficient algorithms (e.g., square-root Kalman filter), and to employ a variety of applications [2, 7, 29, 34, 73, 84, 111, 133].

1.2.1 Extended Kalman Filter (EKF) and Other Nonlinear Filters

The Kalman filter was originally formulated for a linear problem composed of a linear process model and a linear measurement model each of which included additive Gaussian noise[2]. In order to apply the original Kalman filter to nonlinear systems, which are the most commonly occurring cases in real-world systems, Schmidt at the Ames Research Center of NASA discovered the extended Kalman filter which can be applicable to a nonlinear system by linearizing it around the current state estimate [86, 108]. Since then, it has been the most widely used extension for most real-time nonlinear applications of Kalman filtering. The first successful application

[2]The original derivation of the Kalman filter started with more general assumptions [60]. It can be applied to systems with probability densities of non-Gaussian noise while we only maintain the Gaussian components (mean and covariance) of the densities in the recursive estimation. Moreover, the Kalman filter does not assume linearity of the system models while the recursive form of measurement updates is linear and several expectations, such as the expectations of nonlinear process and measurement models and the expectations in the calculation of covariances, remain in general nonlinear forms. In general, these expectations can only be calculated exactly under the condition of a linear state-space model with Gaussian noise. The Kalman filter can only be an exact solution to the optimal Bayesian recursive estimation problem in these conditions. Nevertheless, this fact does not imply the Kalman filter can not be applied to nonlinear, non-Gaussian systems, but it still can be the optimal Gaussian approximate linear estimator to these systems once the assumptions are satisfied. Refer to [115 pp.28-30] for more details.

was on the trajectory estimation problem for the Apollo project, a manned mission to the Moon and back [34, 86, 108].

Even though the extended Kalman filter is still the most widely used version because of its rather intuitive and easy extended implementation, it has its own limitations in two perspectives. Firstly, its estimation performance is strongly dependent on the nonlinearity of the problem at hand. In the formulation of the extended Kalman filter, the algorithm of the linear Kalman filter is directly applied to the linearized approximation of the nonlinear state transition dynamics and nonlinear observation map where the approximate state transition matrix and measurement matrix are provided by taking the first-order Jacobian matrices in each Taylor series expansion of the nonlinear models around the current state estimates. For this reason, if the nonlinearities in the equations that describe the physical system are significant, the filter can be considerably inaccurate and divergent. Another limitation of the EKF is that it can be applied only to systems with Gaussian noise. If the noise in the process model and/or the measurement model is non-Gaussian, the EKF cannot be applied.

In order to overcome these two major limitations of the EKF, numerous research has been performed on developing new nonlinear filters. These include second-order nonlinear filters, Monte-Carlo simulation filters, Gaussian sum filters, grid-based filters, Benes and Daum filters, unscented Kalman filters, and particle filters to name a few. As a first natural extension of the EKF, second-order nonlinear filters use the second-order Taylor series expansion for approximating the nonlinear process and observation models. The expanded second-order approximations are directly applied to the original Kalman filter algorithm, similarly in the formulation of the EKF [84, 112]. As a second extension, Monte-Carlo simulation filters are a combination of the extended Kalman filter and Monte Carlo stochastic simulations, in which each expectation of a nonlinear function is evaluated by random draws [112]. Thirdly, the Gaussian sum filters are an extension of the EKF in the perspective of

probability density approximation. In this filter, posterior density is approximated by a Gaussian mixture model that is a finite linear mixture of Gaussian densities or a weighted sum of Gaussian density functions. Each Gaussian density component of the Gaussian mixture model can be propagated using a separate KF or EKF. This filter is an especially reasonable approach when the posterior density is multimodal, but it still has some problems including the recursive formulation of weights on each density [100, 112]. Another extension is the grid-based filter, sometimes referred to as a (direct) numerical integration filter, in which each probability density function of the optimal Bayesian recursion integrals are approximated by direct numerical integrations in the form of large but finite sums over uniform grids or nodes that tile the complete area of interest. These filters become too numerically expensive to be used practically even when the dimension of state space goes moderate [100,112,115]. Another important class of nonlinear filters is finite-dimensional filters (or Benes and Daum filters), which can be applicable to a special class of nonlinear dynamic systems (assuming a linear measurement equation) as exact optimal nonlinear recursive filters. Benes first formulated this kind of filter and later Daum extended it to various forms of filters [18,100]. Two of the most successful streams of recent research in the area of nonlinear filtering are unscented Kalman filters as a Gaussian approximation method and particle filters as a sequential Monte Carlo method. These two will discussed in detail in the subsequent sections and chapters.

1.2.2 Unscented Kalman Filters and Derivative-Free Filters

The unscented Kalman filter (UKF), developed by Julier et al. [53, 55, 56, 58, 59], is a new type of Kalman filter that is applicable to Gaussian-noise, nonlinear estimation problems. It falls into the category of a Kalman filter in the sense that filter updates are performed in the same framework as that of the Kalman filter, that is, both the mean state estimates and the error covariance estimates are updated in two

separate steps consisting of a prediction step (or time update step) and a correction step (or measurement update step). On the other hand, the UKF is a new filter compared to the Kalman filter in the sense that it does not use linearization of nonlinear functions in order for the UKF to be applied to nonlinear estimation problems, and thus does not require the computation of Jacobian matrices that involves significant implementation difficulties in most applications. The propagation of mean state estimates and error covariance estimates through nonlinear functions employs the *unscented transformation* [56]. This transformation is a method for calculating the transformed statistics of a random variable that undergoes a nonlinear transformation, for example, the mean and the covariance of the nonlinear function of a random variable. The basic idea behind the development of the unscented transformation is that *approximating a Gaussian distribution is easier than approximating an arbitrary nonlinear function or transformation* [58]. In order to more exactly capture the statistics through a nonlinear transformation, a set of deterministically chosen weighted sample points (referred to as sigma points) is used to parameterize the mean and the covariance of the probability distribution.

Researchers of the UKF argue that two main advantages of the UKF over the EKF are its accurate estimation and its easy implementation. When it comes to estimation performance, the UKF provides accuracy comparable to second-order filters without Jacobian or Hessian matrix computations. It employs the unscented transformation to deterministically chosen sigma points for the propagation of Gaussian statistics instead of using the linearization of nonlinear functions (e.g., nonlinear process and observation models). In terms of implementation, the UKF is easier to implement than the EKF. First, the UKF does not necessitate the computation of Jacobian matrices whose analytical expressions are often very difficult to derive. Second, the UKF algorithm is very generic compared to the EKF since it reduces model dependent parts such as Jacobian matrices, allowing the process and observation models to be

considered as black boxes in this framework [18, 100, 115, 124].

Despite the advantages of the UKF, it may require some considerations depending on each application at hand mainly because of the formulation differences between the UKF and the EKF. The first one is about computational cost. Even though the developers of the UKF argue that the computational cost of the UKF is comparable to that of the EKF, it can be more expensive than that of the EKF depending on the specific problem at hand. Since the UKF formulation replaces a function call and Jacobian matrix computation (e.g., one function call and a slope at the current point in the one-dimensional case) by several function calls at every sigma point (e.g., three function calls at the current point and two perturbed points around the current point in the one-dimensional case), the computational cost of the UKF and the EKF may be dependent on those of the Jacobian matrix computations and the function calls in each specific application. Secondly, we need to take care of the case in which state components are not totally independent but related to each other through some constraints. Since the UKF formulation is based on function calls at perturbed states around the current estimated state, the perturbed states (or sigma points) must be determined while the constraints among the state components are maintained.

Since the introduction of the standard UKF in the mid 1990's, considerable research has been performed on the extension of the original formulation and the capability of various applications. The developers of the UKF presented a series of research results on the different effective forms of the unscented transformation. First, the scaled unscented transformation is suggested to guarantee at least second-order accuracy in computing the mean and covariance. In this research, a sonar sensor in a mobile robot detects a beacon in its environment and returns polar information (range and bearing). This needs to be converted to estimate the position in Cartesian coordinates. The effectiveness of the scaling is illustrated by considering conversions

from polar to Cartesian coordinates with large angular uncertainties [50]. This problem is closely related to our vision-based tracking problem that will be described in detail in related chapters. The effectiveness or accuracy in handling the coordinate conversions from polar to Cartesian coordinates provides a motive for using the UKF in our vision-based tracking problem since our problem is, in a sense, a generalized problem of this simple case (estimating positions and velocities in Cartesian coordinates from the measurements of one subtended angle and two bearings). As an effort to minimize the number of sigma points that determine computational cost, Julier et al. [54] presented a method in which a set of $n + 1$ sigma points (for n-dimensional state space systems) can fully capture all of the known statistics of the mean and the covariance. As an extension of this research, Julier [51] derived an algorithm to compute a set of $n+2$ sigma points that is the minimal skew set of simplex points and thus minimizes the magnitude of the third-order moments. The UKF can also provide a clear framework for some sophisticated problems. Julier et al. [56,57] presented methods of how to deal with assignment ambiguity for the problem of many uncertain landmark-based navigation systems and how to deal with uncertain time-delayed measurements, both of which are based on the covariance union (CU) algorithm.

A number of other researchers have also been interested in related subjects. Lefebvre et al. [59,70] argue that the UKF falls into the category of linear regression Kalman filters because it is based on statistical linearization [29, 84] rather than analytical linearization (as the EKF). Some different derivative-free nonlinear filters in this category have also been developed by several other researchers including Schei [106], Ito et al. [41], Nogaard et al. [90, 91], and Lee et al. [69]. Research on a higher-order unscented transformation was conducted by Tenne et al. [113], who proposed a higher-order unscented filter (HOUF) that can capture the statistics up to the third order. Levesque [72] proposed a technique that matches the moments of the probability density function up to the sixth order with a set of $2n + 3$ second-order

simplex sigma points. Merwe et al. [115–119, 123, 124] also researched extensively about both the theoretical aspects and various applications in a more general framework of sigma-point Kalman filtering, for example, dual estimation (state and parameter estimation) [117,123], the square-root UKF [116,117], the unscented particle filter [118,124], and integrated navigation systems [115,119]. Some researchers have attempted to apply the UKF to several important aerospace applications including integrated navigation systems [14, 25, 67, 92, 115, 119, 128], vision-based navigation systems [67,68], and target tracking problems [100,131].

1.2.3 Sequential Monte Carlo Estimation Methods (Particle Filters)

Even though the UKF can provide better estimation performance in many relatively higher-order nonlinear dynamic systems and observation maps compared to the EKF, it still uses the Gaussian noise approximation, which is not appropriate for application to some highly nonlinear non-Gaussian noise problems. For example, many target tracking problems and image tracking problems fall into the category of nonlinear estimation problems with multi-modal or other complex probability distributions [23, 100]. While the Gaussian sum filter can handle more or less these problems by approximating the posterior pdf with a finite Gaussian mixture (or a weighted sum of Gaussian density functions), it can be effectively applicable to a very limited range of problems. As an approach to surmount the shortcomings (e.g., Gaussian noise assumption for a posterior pdf, limited applicability to relatively modest nonlinear systems) of other estimation methods, one important recent development in nonlinear estimation theory is the sequential Monte Carlo methods (or referred to be as particle filters). The sequential Monte Carlo methods draw part of their foundation from the Monte Carlo sampling methods whose development goes back to the 1940's and 1950's when their modern formulation was shaped in the Los Alamos Laboratory and applied to the problems of physical sciences [36]. The advantage of the Monte Carlo

sampling methods is that they can numerically tackle very complex, analytically intractable integration problems [81, 102]. The other side of the sequential Monte Carlo methods comes from the recursive Bayesian statistics that allows sequential (or recursive) estimation without the necessity of saving all previous information such as state and observation history [38]. In fact, the sequential Monte Carlo methods provide an attractive online estimation approach by implementing the Monte Carlo sampling methods in the recursive Bayesian estimation framework. Specifically, the sequential Monte Carlo methods recursively update the posterior density functions in the Bayesian estimation framework in which the posterior density functions are directly approximated by a set of weighted samples (or particles) based on Monte Carlo integration methods. Since these methods do not make any assumption on the form of the posterior density functions, they can be applied to a general class of nonlinear, non-Gaussian dynamic systems.

In the conceptual solution of general Monte Carlo integration methods, a set of weighted samples (or particles), drawn directly from the true posterior distribution, is supposed to approximate the integrals of posterior density functions and consequently to approximate the integrals of expectations to discrete finite sums. However, in most nonlinear estimation problems, it is rarely possible to effectively generate samples directly from the true posterior distribution, which may be multi-variate, complex-shaped, and not exactly known but only to a proportional constant. To circumvent the difficulty of direct sampling from the true posterior density functions, general Monte Carlo integration methods typically apply the importance sampling method in which samples are drawn from easier-to-implement importance (or proposal) density functions rather than the unknown or hard-to-sample true posterior density functions.

Once the importance sampling concept for the general Monte Carlo integration methods is formulated in the recursive Bayesian estimation framework, it results in

the sequential importance sampling (SIS) algorithm that constitutes the fundamental skeleton of most sequential Monte Carlo methods developed over the last several decades. These sequential Monte Carlo methods are collectively referred to as particle filters in which a number of independent random state samples (called particles), drawn from the proposal density functions and thus representing the posterior distributions, and their associated weights are recursively updated by incorporating the state transition model, the observation model, and current measurements in the recursive Bayesian estimation framework. These SIS-based sequential Monte Carlo methods suffer from the well-known defect, the sampling degeneracy phenomenon, which means that the normalized weights of all particles (or samples) except one become negligible as time increases (or recursive steps progress). By introducing an additional sampling step (resampling step), Gordon et al. [33] discovered a new type of particle filter that could overcome the sampling degeneracy problem by employing the sampling importance resampling (SIR) algorithm. Due to the development of this effective algorithm and the increase in recent computing power, the particle filter has become a practically implementable approach applicable to many difficult nonlinear estimation problems. Recently, a significant amount of research activities have been performed to extend the capability of basic particle filters in both theoretical aspects and real-world applications [18, 19, 23, 33, 100, 115]. In the theoretical perspectives, many researchers have worked on developing new theories to improve the performance of basic particle filters. Some impressive results are the bootstrap filter (or SIR filter), the auxiliary particle filter (or auxiliary SIR filter), the regularized particle filter, the Markov chain Monte Carlo (MCMC) particle filter, the local linearization particle filter, the Gaussian particle filter, the Kernel particle filter, the multiple model particle filter, the unscented particle filter, and the marginalized particle filter or Rao-Blackwellized particle filter, to name a few [23, 100, 115]. Meanwhile, many have

also focused on demonstrating the filter's capability when applied to a variety of real-world applications including bearings-only tracking, mobile robot localization, visual shape and motion learning, visual tracking, terrain-aided positioning and navigation, GPS/INS integration, spacecraft attitude estimation, and target tracking [23, 100].

1.3 Integrated Navigation System

Inertial navigation is the process of estimating ownship states (e.g., the position and velocity of a vehicle in a chosen reference) by using a set of onboard accelerometers and gyros. Inertial navigation systems (INS) are categorized into two types: gimballed and strapdown [62, 63, 80]. Since the development of the first operational INS installed in the German V-2 rockets of World War II, gimballed INS have been the earlier type of INS. In the gimballed INS, the gimballed platform to which the three-axis accelerometers and gyros are fixed is isolated from the rotations of the vehicle. The directions of the three accelerometers are aligned to a specific orientation (e.g., north-east-down direction) relative to an inertial space, and the three integrating gyros that provide outputs proportional to a rotation angle are used to maintain the gimbal direction by the mechanism of nullifying the gimbal rotation with respect to a constant inertial orientation. In this gimballed INS, the inertial position and velocity of a vehicle can be obtained, at least conceptually, just by integrating the three-axis accelerations measured from the accelerometer triad aligned in the inertial reference frame. In a practical point of view, since the accelerometer does not measure directly acceleration but rather measures specific force that results from the addition of acceleration and gravity, we need a gravity model in order to calculate the acceleration. The gravity model may be very sophisticated depending on the required accuracy of the involved navigation systems. Since these gimballed INS can provide very accurate, self-contained navigation solutions even with not so delicate gyros and accelerometers, they have been prevalently used in most military

and commercial vehicles including ballistic missiles, fighter aircraft, naval ships, and airliners. On the other hand, the gimballed INS has many disadvantages. Because of the moving gimballed mechanism, it is very complex and bulky, includes delicate devices such as sliprings, and is thus difficult and time-consuming for maintenance. Above all, it is very expensive.

In the early 1970's when the gimballed INS technology was fully matured, INS researchers started working on a simpler INS mechanism (referred to as the strapdown INS) in which the onboard accelerometers and gyros were fixed (or strapped down) to a vehicle so that the accelerometer triad measures three-axis accelerations in a body-fixed frame and the rate gyro triad measures three-axis rotation rates of the body frame. The angular rates are integrated to trace the instantaneous orientations of the vehicle or those of the accelerometers. At the same time, the three-axis accelerations in the body-fixed frame are transformed to reach the accelerations in the inertial (or navigation) frame, and the transformed inertial accelerations are integrated to obtain the position and velocity in the inertial space. Similar to the case of the gimballed INS, a gravity model is typically utilized in determining the gravitational accelerations from the specific forces measured by the accelerometers. Two basic technologies behind the development of the strapdown INS were, on the one side, the advent of high-speed digital computers and, on the other side, the invention of high-performance rate gyros. In order to keep track of the instantaneous attitude vector (e.g., Euler angles or quaternions) and the instantaneous coordinate transformation matrix from the body-fixed frame to the navigation frame, we need to first update the attitude vector at every moment by numerically integrating the attitude differential equations and then compute the coordinate transformation matrix by using the current attitude vector. This rather computationally costly process could be performed at very high rates as a result of the progress in powerful onboard computers. In addition, the invention of more reliable rate gyros including ring laser

gyros (RLG) might be the most significant factor for the maturation of the strapdown INS since they could provide larger dynamic range and more accurate angular rate measurements. Even though both types of INS are still widely used as standard equipment in almost all commercial and military air vehicles, the strapdown INS have been gradually replacing the gimballed INS mainly because of the several advantages including the compact size, easier maintenance, cost effectiveness, and reasonable accuracy especially when they are aided by some external navigation devices such as the global positioning system (GPS), long-range navigation (LORAN), and tactical air navigation (TACAN) [62, 80, 107].

The strapdown INS can provide high-rate ownship states including the vehicle inertial position, inertial velocity, and attitude information necessary for a flight GNC system. The measured three-axis vehicle accelerations and angular rates from accelerometer and rate gyro triads, respectively, are integrated to obtain the vehicle inertial position, inertial velocity, and attitude in an onboard computer. Even until recently, the inertial navigation systems, which provide enough accuracy through delicate devices and have usually been installed in military or commercial long-range navigation systems, have been considered to be very expensive devices. However, due to the easy availability of low-cost inertial measurement units (IMU) consisting of accelerometer and rate gyro triads rigidly mounted on a vehicle, the strapdown INS has become a backbone in cost-effective autonomous unmanned aerial vehicles [4, 83].

Even though a strapdown INS equipped with a low-cost MEMS-based IMU can provide necessary high-rate ownship state information for a GNC system of a UAV, the accuracy in the vehicle position, velocity, and attitude information rapidly degrades with time since measured vehicle accelerations from accelerometers and angular rates from rate gyros are generally susceptible to various measurement noise sources [5, 7, 62, 103, 114]. Attempting to increase the navigation accuracy by incorporating more accurate IMU measurement devices generally causes the INS system

to become very expensive. On the other hand, recent research has shown that the growth of numerical errors in IMU navigation with time can be effectively prevented by using low-cost absolute aiding sensors such as global positioning system (GPS), magnetometer, altimeter, and so on [21,64,67,94,96,119,120,129,132]. By combining the low-cost IMU with absolute aiding sensors, relatively accurate state estimation information can be provided to low-cost autonomous small UAV navigation systems. This integrated navigation system solves the time-degrading accuracy problem of a low-cost strapdown INS by combining the short-term high-rate data characteristics of the low-cost INS with the low-rate but relatively time-independent accuracy of absolute aiding sensors.

Some recent research on integrated navigation systems has been applied to the navigation systems of autonomous UAV's. Eck and Geering [24] presented an augmented INS/GPS error dynamics model, capable of being fitted to the linear Kalman filter, for an autonomous UAV helicopter. The error model includes not only the nominal INS state errors but also the inertial sensor errors and the GPS errors. Wenger and Gebre-Egziabher [129] discussed a multi-sensor integrated navigation system that fuses information from a low-cost IMU, a magnetometer triad, and a GPS receiver. The vehicle position from the GPS receiver and the yaw (or heading) angle from the magnetometer are used as the aiding measurements for the EKF-based navigation system, and the performance simulations of the navigation system for UAV applications are also presented in the paper. Vasconcelos et al. [120] proposed an aiding technique in which a low-cost strapdown INS is aided by GPS position measurements and additional vector measurements with an application to UAV's. In this paper, GPS position, magnetic vector, and gravity vector measurements are fused with strapdown INS measurements in the framework of the EKF. The magnetic vector measurements from the magnetometer triad and the gravity vector from the acceleromter triad are used to improve attitude observability. Winkler et al. [130] suggested a technique in

which a tightly-coupled GPS/INS is aided by an onboard camera for horizontal attitude detection. With an application to autonomous mini and micro aerial vehicles, the multi-sensor data fusion is performed in the framework of the linear Kalman filter with an error state vector and a sequential processing of single GPS measurements. Additional important recent efforts are the attempts to apply the new nonlinear filtering methods to improve the performance of the integrated navigation systems [92]. R. van der Merwe and E. A. Wan performed research on improving the EKF-based navigation system of the MIT X-Cell-90 helicopter by incorporating the sigma point Kalman filter (SPKF) [115, 119]. By using latency-compensated measurements from the GPS position/velocity and an altitude measurement from the barometric altimeter as the aiding sensor measurements, they could exhibit the performance improvement in the ownship state estimation. El-Sheimy et al. [25] presented research results on the performance comparison of the integrated GPS and MEMS-based INS by using both the EKF and the UKF. They showed some benefits of using the UKF including the case of handling large initial attitude errors. Wendel et al. [128] also compared the performance of the EKF and SPKF for a tightly-coupled GPS/INS navigation system. Their Monte Carlo-like simulations averaged over 25 runs showed nearly identical performance in their specific problems except for some unrealistic large initial position errors.

Even though some researchers attempted to apply particle filters to integrated GPS/INS navigation systems, no effort produced conclusive result [11, 31, 32, 89]. One of the objectives in this research is to develop an integrated navigation system that is composed of a MEMS-based low-cost IMU and several aiding sensors including a differential GPS receiver, a magnetometer, and an altimeter in the accurate UKF framework. The aiding sensor measurements that have different vector sizes and data rates are effectively fused with IMU measurements in the novel UKF with sequential measurement updates.

1.4 Nonlinear Filtering and Applications to Target Tracking Problems

Target tracking is one of the most important areas in nonlinear filtering applications. The problem intrinsically needs help from the filtering tools including the Kalman filters and particle filters. As the sensing mechanism for detecting and tracking flying targets, two major devices have traditionally been a radar and a seeker. A myriad of research results have piled up in this area. The visual tracking system is a recent promising addition. Since a thorough survey on this subject is difficult, only a small portion of recent attempts will be reviewed regarding the current work.

1.4.1 Bearings-Only Tracking

The vision-based tracking problem in current work is, in a sense, an extended version of a typical bearings-only problem [16, 71, 101]. The typical bearings-only problem, sometimes referred to as "target motion analysis" (TMA), is the estimation of relative kinematics, such as the relative position and velocity of a moving target in a planar motion, using only noise-corrupted bearing measurements. Our problem, on the other hand, attempts to estimate not only relative kinematics such as relative position and velocity in three-dimensional space but also target characteristics such as target size and target acceleration components using two bearing angles and a subtended angle. Due to a variety of important practical applications, even the relatively simple bearings-only problem has resulted in numerous research efforts in order to overcome the difficulty involved in inherent nonlinearity and observability issues.

In their celebrated work, Aidala and Hammel [1] could reformulate the bearings-only TMA estimation problem in decoupled form by introducing the novel modified polar (MP) coordinates since the MP coordinates decouple observable and unobservable components of the estimated state vector. The reformulated problem in the

MP coordinates could be successfully solved by the EKF and thus overcame the defects of the EKF since the attempt of employing the EKF to deal with the original bearings-only problem in Cartesian coordinates often leads to filter instability and asymptotically biased results. The state decoupling by the MP coordinates is known to be the mechanism that provides filter stability since it prevents covariance matrix ill-conditioning. Furthermore, the MP state estimates are asymptotically unbiased. Grossman [35] proposed an approach for the bearings-only tracking problem by using a hybrid coordinate system in the EKF framework. The hybrid coordinate approach combines linear propagation of the state and error covariance in the Cartesian system with a linear update in the MP coordinates. Peach [97] presented the research results in which the typical two-dimensional bearings-only tracking problem is solved by three approaches: the Cartesian EKF, the MP EKF, and the new range-parameterized EKF. Comparisons show considerably better tracking performance with the range-parameterized EKF. Lerro and Bar-Shalom [71] proposed a new type of Cartesian EKF (referred to as the bias compensated Cartesian filter (BCCF)) that effectively solves the bearings-only target state estimation problem in the Cartesian coordinates. This filter incorporates a new mechanism for estimating the range bias to compensate the position estimates in the traditional Cartesian filter. Cui and Zhu [16] discussed the real-time implementation of the bearings-only target tracking system. With the MP coordinates, a state and asymptotically unbiased EKF is implemented by using a covariance square-root filtering and fading memory filtering with variable coefficients.

Recent attempts are mostly focused on applying nonlinear filters to this problem. Xu and Liping [131] attacked this problem by introducing the UKF and compared their results with those obtained by the EKF. The RMS position and velocity error by the UKF show better estimation performance as time progresses. Gordon et al. [33] used this famous problem as one of the applications that shows the effectiveness of their historical resampling-based particle filter (referred to be as the bootstrap

filter or SIR filter). Carpenter et al. [10] also solved this problem in the framework of particle filtering in which a stratified sampling method is applied instead of the resampling method. Ristic et al. [101] intensively studied this typical bearings-only target tracking problem in terms of coordinate systems, target maneuverability, and a variety of nonlinear filters.

There are also some researchers who are focusing on the extended applications of the typical bearings-only problem. Kwok et al. [66] worked on developing the path planning algorithm for bearings-only simultaneous localization and mapping (SLAM) based on the EKF. Karlsson and Gustafsson [61] studied several realistic bearings-only applications such as air-to-air passive ranging, air-to-sea applications with terrain induced constraints, and sea-to-sea tracking with a passive sonar sensor. As the filtering tools, the SIR-based particle filter and marginalized particle filter were compared to the filter bank method using a range-parameterized EKF. Brehard and Cadre [6] presented research results on multi-sensor data fusion architectures based on Bayesian estimation. Target tracking for a single target using distributed multi-sensor bearing measurements are addressed in their research.

1.4.2 Maneuvering Target Tracking

Li and Jilkov provided a series of comprehensive surveys on maneuvering target tracking. In Ref. [77], a variety of target motion models and target tracking kinematic models are presented. Ref. [74] presents dynamic motion models and target tracking kinematics models for ballistic targets. Ref. [75] provides several measurement models in various coordinate systems. It also furnishes several pseudomeasurement models to deal with the bias problems of the EKF or kinematic constraints. Ref. [76] reviews a number of decision-based techniques for maneuvering target tracking. Ref. [79] surveys on multiple model (MM) methods for maneuvering target tracking. MM methods

are considered to be the most reliable methods especially when the maneuvering target motion is uncertain or nonlinear. Several approximation techniques for nonlinear filtering are presented in Ref. [78] including EKF's, second-order filters (e.g., Gaussian second-order filter), and unscented Kalman filers. Mazor et al. [85] provided a survey on interactive multiple model methods in target tracking. Farina et al. [26] studied the tracking problem of a ballistic target by comparing several nonlinear filters such as the EKF, the UKF, and the particle filter. Minvielle [88] also presented a general review on the ballistic vehicle tracking problem in the framework of several filtering methods including the EKF and particle filters. Bruno [8] presented two improved particle filter algorithms for ballistic target tracking. The improved SIR particle filter and the auxiliary particle filter are tested for typical target tracking with tracking radar measurements composed of one range and one angle quantity. Ref. 23 illustrates the particle filter for dealing with the problem of tracking a maneuvering target whose maneuvering modes may be highly uncertain, for example, straight, left turn, or right turn. The problems of multiple target tracking were also effectively handled in the Bayesian filtering framework [40, 95, 121].

1.5 Vision-Based Tracking System

Recently, one of the most attractive tracking sensor systems has been the vision sensor system. This vision sensor system is typically a combination of hardware equipment (e.g., video camera, camera-image capturing hardware or frame-grabber to capture still image from video stream, onboard computer to process the computer vision algorithms) and computer vision software (e.g., captured-image processing software to extract target image features) [42]. The vision system has played a very important role in many applications as one of the key elements determining guidance and control commands [43, 65, 98, 99, 125, 127]. The vision sensor is especially useful when the mission of UAV's is related to the tracking of other moving objects or the awareness

of the global environment in which they fly [122, 125, 126]. In fact, most of the recent UAV missions require both capabilities.

The first major objective of the vision sensor is its use as a tracking sensor system for measuring target tracking states. Tracking is the estimation of the state of a moving object or target based on remote measurements using one or more sensors at fixed locations or on moving platforms. Restricting the objective of tracking applications to aerial vehicles, two typical tracking sensors for aerial target tracking are seeker and radar systems that are usually very expensive and sometimes bulky. Moreover, the associated technologies of these systems may not be easily accessible since they are classified as military technologies. In contrast, the use of a vision sensor as a tracking sensor of low-cost UAV's is very attractive since it is a very efficient sensor due to its compact size and reduced cost while providing rich information. Furthermore, it is a very natural choice in that many living creatures including humans, birds, and insects depend their motion or navigation primarily on their eyes (or vision sensors).

The other indispensable capability for many autonomous UAV missions is determining the global position and velocity of target objects in the world. The knowledge of this target-state information allows the autonomous UAV's to operate safely in a cluttered environment with obstacles, buildings, terrain, and other vehicles. This knowledge also becomes an essential element for the successful completion of complex mission requirements. Target states in the global coordinate system can be obtained by adding the ownship states and the target-tracking states as shown in Figure 1.4. In other words, the estimation of target global motion can be achieved by combining ownship global motion (or ownship states) calculated by the ownship integrated navigation system and target-ownship relative motion (or target tracking states) as processed by the vision-based target tracking system.

1.6 The Outline of Thesis

Chapter 2 summarizes various nonlinear filtering algorithms that include the newly developed UKF with sequential measurement updates and the extended marginalized particle filter as well as the reference EKF with sequential measurement updates and the standard UKF. Chapter 3 presents the detailed algorithms of the integrated navigation system that utilizes both the EKF with sequential measurement updates and the UKF with sequential measurement updates. Chapter 4 provides a new vision-based target tracking system that uses the unscented Kalman filter. Chapter 5 suggests an innovative vision-based tracking system based on the new extended marginalized particle filter. In this framework, non-Gaussian vision measurements may be easily incorporated as a probabilistic input to the vision-based tracking system. Chapter 6 summarizes current contributions and recommended future research.

CHAPTER 2

NONLINEAR ESTIMATION

2.1 Introduction

This chapter presents several nonlinear filters that include standard and newly developed filters. Section 2.2 introduces some mathematical concepts and tools about probability and statistics which are used in the subsequent sections. Section 2.3 summarizes the standard EKF algorithm and an EKF algorithm with sequential measurement updates. The EKF with sequential measurement updates becomes the reference algorithm for the design of the baseline integrated navigation system that provides efficient framework for fusing various sources of multi-rate sensor information. Section 2.4 introduces the unscented transformation leading to the standard UKF algorithm and a new UKF algorithm with sequential measurement updates. The standard UKF is applied to the design of a new vision-based tracking system, and the new UKF with sequential measurement updates is used to design an efficient integrated navigation system that combines the advantages of the UKF design and sequential measurement updates. Section 2.5 summarizes the concept of particle filtering which becomes the stepping stone in the development of a new extended marginalized particle filter presented in detail in Section 2.6. This new filter can be applied to systems that include non-Gaussian measurements and internal nonlinear dynamics with Gaussian process noise characteristics. In this framework, a new vision-based tracking system will be designed and tested with image measurements obtained in flight tests or high-fidelity 6-DOF (degree of freedom) simulations. This framework can be easily extended to incorporate non-Gaussian image measurements as the input to the tracking system.

2.2 Mathematical Preliminaries

Bayesian theory, named after its British developer Thomas Bayes, first appeared in 1763 and has become one of the most important branches in statistical inference since the French mathematician Pierre-Simon de Laplace formulated the famous modern form of the *Bayes rule* [17]. The theory describes the fundamental probability law governing the process of logical inference and makes it possible to model the uncertainty of the real world and the outcomes of interest by incorporating prior knowledge and observational evidence. A few examples of the many application areas of Bayesian inference include statistical estimation or filtering (referred to as Bayesian estimation or Bayesian filtering), statistical decision-making, and machine learning (Bayesian learning). If we use the Bayes rule in the state estimation of a dynamic system expressed in state-space form, we can derive the formulation of the recursive Bayesian estimation that provides the most general solution framework for dynamic state estimation problems. For example, the celebrated Kalman filter is the optimal analytical solution to this recursive Bayesian state estimation in a linear Gaussian-noise case, and the extended Kalman filter and the unscented Kalman filter are analytically approximated suboptimal nonlinear filters applied to nonlinear systems with Gaussian noise. Moreover, the particle filter is a way of sequential Monte Carlo estimation based on this recursive Bayesian formulation. The particle filter is applicable to nonlinear, non-Gaussian systems.

A. Dynamic State-Space Models

In stochastic estimation problems, the dynamical system is usually expressed in the following discrete-time state-space form:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \tag{2.1}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k), \tag{2.2}$$

where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k$, and $\mathbf{u}_k$ are the state, observation, and control input vectors,

respectively, at discrete time $t = t_k$; $\mathbf{w}_k$ and $\mathbf{v}_k$ are process and measurement noise sequences, respectively; and $\mathbf{f}_k$ and $\mathbf{h}_k$ are some known vector-valued (possibly nonlinear time-varying) functions. In addition, $k \in \mathbb{N}$ is the time index, $n \in \mathbb{N}$ is the dimension of the state vector, and $\mathbb{R}$ and $\mathbb{N}$ are the set of real numbers and natural numbers, respectively. Here, the first equation is referred to as the process model or state transition (or evolution) equation, and the second as the measurement model or the observation map. The process model characterizes the state transition probability $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ that is related to the process noise sequence $\mathbf{w}_k$, and the measurement model determines the likelihood density $p(\mathbf{y}_k|\mathbf{x}_k)$ that depends on the measurement noise sequence $\mathbf{v}_k$. Such a discrete-time state-space model is generally obtained by discretizing the following continuous-time state-space model:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \tag{2.3}$$

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{v}(t)), \tag{2.4}$$

where $\mathbf{x}_k = \mathbf{x}(t_k)$. In the discretization process, we generally assume that $\mathbf{y}_k = \mathbf{y}(t_k)$, $\mathbf{v}_k = \mathbf{v}(t_k)$, and $\mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) = \mathbf{h}(t_k, \mathbf{x}(t_k), \mathbf{v}(t_k))$, but we need to be careful that $\mathbf{w}_k \neq \mathbf{w}(t_k)$ and $\mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \neq \mathbf{f}(t_k, \mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{w}(t_k))$. Sometimes, it is more pertinent to express a dynamic system in the following mixed-time state-space model:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \tag{2.5}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k), \tag{2.6}$$

since usual real-world dynamic systems can be more accurately modeled in a continuous-time domain (as differential equations), and sensor measurements are usually available at discrete-time instants. The process model expressed in the continuous-time differential equation can be discretized into a discrete-time difference equation by using a first-order Euler algorithm (or by "zero-order holds" sampling) or higher-order algorithms, depending on the necessary accuracy. Throughout this work, the last

mixed-time state-space model is preferably utilized with the additional discussion of implemented discretization methods even though most of the filtering theories are discussed based on the discrete-time model.

B. Bayesian Statistics and Stochastic Estimation Problem

Bayesian theory is a branch of mathematical probability theory and becomes an important element for the development of stochastic estimation theory using Bayesian inference. The following are some of Bayesian statistics:

- **Marginal Probability Density**: Given the joint probability density function (pdf) $p(\mathbf{x}, \mathbf{y})$, the marginal pdf is

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y})\, d\mathbf{y}. \tag{2.7}$$

- **Conditional Probability Density**: Given the joint pdf $p(\mathbf{x}, \mathbf{y})$ and the pdf $p(\mathbf{y})$, the conditional pdf $p(\mathbf{x}|\mathbf{y})$ (i.e., the probability density of $\mathbf{x}$ given $\mathbf{y}$) is

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} \quad \text{or} \quad p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})\, p(\mathbf{y}). \tag{2.8}$$

- **Bayes Rule**: Given the prior $p(\mathbf{x})$ and likelihood $p(\mathbf{y}|\mathbf{x})$, the posterior $p(\mathbf{x}|\mathbf{y})$ can be obtained by the product of prior and likelihood divided by a normalizing constant as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})}{\int p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}}. \tag{2.9}$$

- **Expectation**: Given the conditional pdf $p(\mathbf{x}|\mathbf{y})$ or the posterior pdf, some averaged statistics of interest can be calculated as

$$\mathbb{E}_{p(\mathbf{x}|\mathbf{y})}[\mathbf{g}(\mathbf{x})] = \int \mathbf{g}(\mathbf{x})\, p(\mathbf{x}|\mathbf{y})\, d\mathbf{x}. \tag{2.10}$$

With the ingredients of Bayesian statistics and dynamic system models expressed in the state-space form, we can now describe the stochastic estimation problem. The

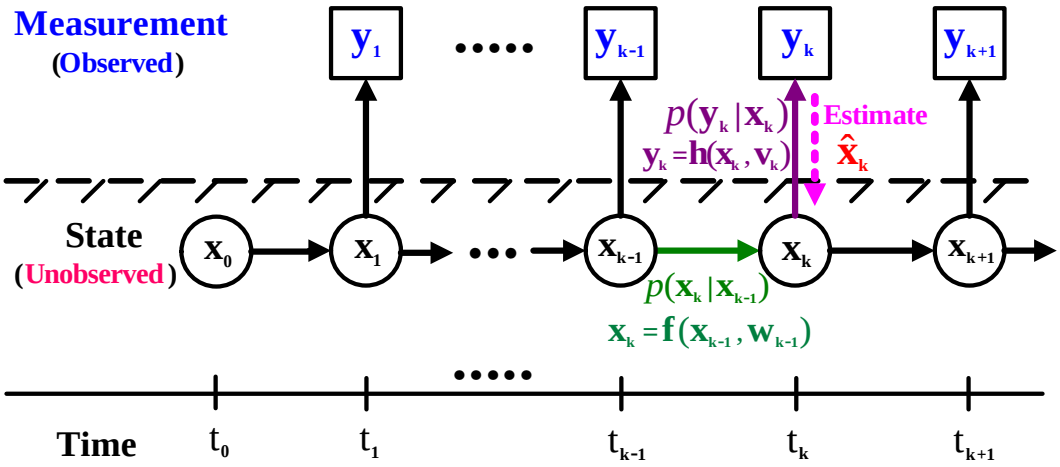


Figure 2.1: Graphic Representation of a Probabilistic Dynamic State-Space Model

generic stochastic estimation problem can be defined as a way of estimating the states of a dynamical system, expressed in time-domain dynamic state-space form, by using uncertain, noisy, and partial observation (or measurement) information. In other words, how to estimate the hidden states $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$ by using the noisy measurements $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$ and the known statistics of the state transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$. The graphical representation of this stochastic estimation problem is described in Figure 2.1. The statistics (or probability density function (pdf)) of the initial state $p(\mathbf{x}_0)$ is assumed to be known.

From the Bayesian statistics, we know that calculating the statistics of interest amounts to determining the posterior pdf because the knowledge of the posterior pdf provides a complete solution to the stochastic estimation problem even though the solution itself is still in intractable integral form. More specifically, given the initial pdf $p(\mathbf{x}_0)$, state transition pdf $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$, estimating the optimal current state at time k based on the sequence of all available measurements $\mathbf{Y}_k \triangleq \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$ up to time k amounts to estimating the posterior pdf $p(\mathbf{x}_k|\mathbf{Y}_k)$ or $p(\mathbf{X}_k|\mathbf{Y}_k)$. Here, $p(\mathbf{X}_k|\mathbf{Y}_k)$ and $p(\mathbf{x}_k|\mathbf{Y}_k)$ are the joint posterior pdf and the marginal posterior pdf (or filtering pdf) of the state at time k, respectively, in which the

following definitions are used:

$$\mathbf{X}_k \triangleq \mathbf{x}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}, \tag{2.11}$$

$$\mathbf{Y}_k \triangleq \mathbf{y}_{1:k} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}, \tag{2.12}$$

where $\mathbf{X}_k$ is the sequence of all available system states up to time k, and $\mathbf{Y}_k$ is the sequence of all available measurements or observations up to time k. Even though the joint posterior pdf $p(\mathbf{X}_k|\mathbf{Y}_k)$ of the state represents full knowledge about the statistics of interest, the marginal posterior pdf $p(\mathbf{x}_k|\mathbf{Y}_k)$ of the state furnishes the complete solution to a filtering problem in a Bayesian framework. That is, the marginal posterior pdf $p(\mathbf{x}_k|\mathbf{Y}_k)$ of the state is sufficient to calculate any optimal estimate of the state in a Bayesian filtering problem. For example, the optimal filtered state estimate in the minimum mean-squared error (MMSE) sense is given by the conditional mean

$$\hat{\mathbf{x}}_k \triangleq \mathbb{E}[\mathbf{x}_k|\mathbf{Y}_k] = \int \mathbf{x}_k \; p(\mathbf{x}_k|\mathbf{Y}_k) \; d\mathbf{x}_k. \tag{2.13}$$

Similarly, an error covariance matrix as an accuracy measure of the state estimate $\hat{\mathbf{x}}_k$ can also be obtained from $p(\mathbf{x}_k|\mathbf{Y}_k)$ as follows:

$$\mathbf{P}_{x_k} \triangleq \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T |\mathbf{Y}_k] = \int (\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \; p(\mathbf{x}_k|\mathbf{Y}_k) \; d\mathbf{x}_k. \tag{2.14}$$

In general, any expectation of the vector-valued function $\mathbf{g}(\mathbf{x}_k)$ of the state $\mathbf{x}_k$ as an averaged statistics may be expressed by

$$\hat{\mathbf{g}}(\mathbf{x}_k) \triangleq \mathbb{E}[\mathbf{g}(\mathbf{x}_k)|\mathbf{Y}_k] = \mathbb{E}_{p(\mathbf{x}_k|\mathbf{Y}_k)}[\mathbf{g}(\mathbf{x}_k)] = \int \mathbf{g}(\mathbf{x}_k) \; p(\mathbf{x}_k|\mathbf{Y}_k) \; d\mathbf{x}_k. \tag{2.15}$$

C. Recursive Bayesian Estimation

By utilizing Bayes rule, we can derive the recursive Bayesian estimation algorithm for the recursive (or sequential) update of the marginal posterior pdf (or filtering pdf) of the state based on previous time-step information. By the recursive computation of the marginal posterior pdf $p(\mathbf{x}_k|\mathbf{Y}_k)$, we can calculate various online estimates of the system state based on the sequence of all available measurements $\mathbf{Y}_k \triangleq \mathbf{y}_{1:k} =$

$\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$ up to time k. The following two assumptions are used to derive the recursive Bayesian filtering algorithm.

Assumption 2.1:

1. The states are a first-order Markov process $p(\mathbf{x}_k|\mathbf{X}_{k-1}) \triangleq p(\mathbf{x}_k|\mathbf{x}_{0:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$.
   (The current state is dependent only on the state at the previous time-step, not on a past state except the previous time-step state. The current state is not dependent on future observations.)

2. The current observation is dependent only on the current state, not on past observations.
   (Observations are conditionally independent of each other given the state. Observations are not dependent on future states.)

The recursive (or sequential) Bayesian estimation algorithm for the posterior pdf can be derived by using the Bayes rule and the above two assumptions in the following manner:

$$\begin{aligned}
p(\mathbf{x}_k|\mathbf{Y}_k) &= \frac{p(\mathbf{Y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k)}{p(\mathbf{Y}_k)} \quad \text{(by Bayes rule)} \\
&= \frac{p(\mathbf{y}_k, \mathbf{Y}_{k-1}|\mathbf{x}_k)\, p(\mathbf{x}_k)}{p(\mathbf{y}_k, \mathbf{Y}_{k-1})} \quad \text{(by the definition of } \mathbf{Y}_k\text{)} \\
&= \frac{p(\mathbf{y}_k|\mathbf{Y}_{k-1}, \mathbf{x}_k) p(\mathbf{Y}_{k-1}|\mathbf{x}_k)\, p(\mathbf{x}_k)}{p(\mathbf{y}_k|\mathbf{Y}_{k-1}) p(\mathbf{Y}_{k-1})} \quad \text{(by conditionalization)} \\
&= \frac{p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{Y}_{k-1})\, \frac{p(\mathbf{x}_k|\mathbf{Y}_{k-1}) p(\mathbf{Y}_{k-1})}{p(\mathbf{x}_k)}\, p(\mathbf{x}_k)}{p(\mathbf{y}_k|\mathbf{Y}_{k-1}) p(\mathbf{Y}_{k-1})} \quad \text{(by Bayes rule)} \\
&= \frac{p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{Y}_{k-1})\, p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \\
&= \frac{p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \quad \text{(by Assumption 2.1.2)}
\end{aligned} \tag{2.16}$$

or

$$\begin{aligned}
p(\mathbf{x}_k|\mathbf{Y}_k) &= p(\mathbf{x}_k|\mathbf{y}_k, \mathbf{Y}_{k-1}) \\
&= \frac{p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{Y}_{k-1})\, p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \quad \text{(by Bayes rule)} \\
&= \frac{p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})}. \quad \text{(by Assumption 2.1.2)}
\end{aligned} \tag{2.17}$$

As a result, we have the following recursive (or sequential) Bayesian estimation algorithm that recursively updates the posterior pdf by using the prior (or prediction) pdf and likelihood:

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)\; p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \tag{2.18}$$

In this recursive Bayesian estimation algorithm, the posterior pdf $p(\mathbf{x}_k|\mathbf{Y}_k)$ is described by the three terms:

- **Prior pdf**: Given the pdf $p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})$ at time $k-1$, the prediction pdf (or prior pdf) of the state at time k can be obtained by the Chapman-Kolmogorov equation:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Y}_{k-1}) &= \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{Y}_{k-1})\; d\mathbf{x}_{k-1} \quad \text{(by marginalization)} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Y}_{k-1})\; p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})\; d\mathbf{x}_{k-1} \quad \text{(by conditionalization)} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\; p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})\; d\mathbf{x}_{k-1}, \quad \text{(by Assumption 2.1.1)} \end{aligned} \tag{2.19}$$

  where Assumption 2.1.1 is used. The state transitional density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the probabilistic model of the state evolution

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}) &= \int p(\mathbf{x}_k, \mathbf{w}_{k-1}|\mathbf{x}_{k-1})\; d\mathbf{w}_{k-1} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{w}_{k-1})\; p(\mathbf{w}_{k-1}|\mathbf{x}_{k-1})\; d\mathbf{w}_{k-1} \\ &= \int \delta\left(\; \mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})\; \right)\; p(\mathbf{w}_{k-1})\; d\mathbf{w}_{k-1}, \end{aligned} \tag{2.20}$$

  where $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$, $\delta(\cdot)$ is the Dirac delta function, and $p(\mathbf{w}_{k-1}|\mathbf{x}_{k-1}) = p(\mathbf{w}_{k-1})$ is used.

- **Likelihood pdf**: The likelihood pdf is the measurement probability

$$\begin{aligned} p(\mathbf{y}_k|\mathbf{x}_k) &= \int p(\mathbf{y}_k, \mathbf{v}_k|\mathbf{x}_k)\; d\mathbf{v}_k \\ &= \int p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{v}_k)\; p(\mathbf{v}_k|\mathbf{x}_k)\; d\mathbf{v}_k \\ &= \int \delta\left(\; \mathbf{y}_k - \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)\; \right)\; p(\mathbf{v}_k)\; d\mathbf{v}_k, \end{aligned} \tag{2.21}$$

where $\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$ and $p(\mathbf{v}_k|\mathbf{x}_k) = p(\mathbf{v}_k)$ is used.

- **Evidence**: The pdf $p(\mathbf{y}_k|\mathbf{Y}_{k-1})$ is the normalizing constant

$$\begin{aligned} p(\mathbf{y}_k|\mathbf{Y}_{k-1}) &= \int p(\mathbf{y}_k, \mathbf{x}_k|\mathbf{Y}_{k-1})\, d\mathbf{x}_k \\ &= \int p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{Y}_{k-1})\, p(\mathbf{x}_k|\mathbf{Y}_{k-1})\, d\mathbf{x}_k \\ &= \int p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{Y}_{k-1})\, d\mathbf{x}_k, \end{aligned} \tag{2.22}$$

where Assumption 2.1.2 is used.

D. Optimal Bayesian Filtering

Once the posterior pdf $p(\mathbf{x}_k|\mathbf{Y}_k)$ of the state is recursively calculated online, we can compute an optimal state estimate with respect to any criterion that measures the optimality. Some criteria for optimality are as follows:

- **Mean (Optimal MMSE Estimate of State)**: The minimum mean-squared error (MMSE) estimate is to find the conditional mean of $\mathbf{x}_k$:

$$\hat{\mathbf{x}}_k^{MMSE} \triangleq E[\,\mathbf{x}_k|\mathbf{Y}_k\,] = \int \mathbf{x}_k\, p(\mathbf{x}_k|\mathbf{Y}_k)\, d\mathbf{x}_k. \tag{2.23}$$

- **Mode (Optimal MAP Estimate of State)**: The maximum a posteriori (MAP) estimate is to find the mode or maximum of the posterior pdf $p(\mathbf{x}_k|\mathbf{Y}_k)$:

$$\hat{\mathbf{x}}_k^{MAP} \triangleq \arg\max_{\mathbf{x}_k}\ p(\mathbf{x}_k|\mathbf{Y}_k). \tag{2.24}$$

If we use the recursive Bayesian estimation algorithm presented in Eq. 2.18, the original Kalman filter can be derived by using both optimal MMSE sense and optimal MAP sense [115].

2.3 Extended Kalman Filters

As a baseline system to estimate the ownship states, an integrated navigation system, which will be discussed in detail in Chapter 3, is designed by using an extended Kalman filter with sequential measurement updates.

2.3.1 Standard Extended Kalman Filter

This section summarizes the standard extended Kalman filter (EKF) as a reference for the development of the EKF with sequential measurement updates. The general nonlinear continuous-time process model and the nonlinear discrete-time measurement model in state-space form are given by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(\mathbf{x}(t), t)\ \mathbf{w}(t), \tag{2.25}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k, \tag{2.26}$$

where $\mathbf{w}(t)$ is the process noise and $\mathbf{v}(t)$ is the measurement noise. Here, $\mathbf{w}(t)$ is a zero-mean and white Gaussian random process[1]

$$E[\mathbf{w}(t)] = \mathbf{0}, \tag{2.27}$$

$$E[\mathbf{w}(t)\mathbf{w}(\tau)^T] = \mathbf{Q}(t)\delta(t-\tau), \tag{2.28}$$

where $\mathbf{Q}(t)$ is its autocorrelation function that represents the (possibly time-varying) intensity of the white noise. If it is time-invariant, this becomes the power spectral density matrix. $\mathbf{v}_k$ is a zero-mean and white Gaussian random sequence[2]

$$E[\mathbf{v}_k] = \mathbf{0}, \tag{2.29}$$

$$E[\mathbf{v}_k\mathbf{v}_k^T] = \mathbf{R}_k, \tag{2.30}$$

where $\mathbf{R}_k$ is its covariance matrix. The initial conditions ($\mathbf{x}(t_0) \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$) are assumed to be known and have the following form:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}(t_0)],\ \ \mathbf{P}_0 = E[(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)^T]. \tag{2.31}$$

Jacobian matrices of the system dynamics and the measurement model are defined as

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t)},\ \ \mathbf{H}_k = \frac{\partial \mathbf{h}(\mathbf{x}, k)}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}(t_k^-)}. \tag{2.32}$$

[1] This will be denoted simply as $\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t))$.
[2] This will be denoted simply as $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$.

A. Time Update (Prediction Step)

Given the state estimate $\hat{\mathbf{x}}_{k-1} = \hat{\mathbf{x}}(t_{k-1})$ and the error covariance matrix $\mathbf{P}_{k-1} = \mathbf{P}(t_{k-1})$ at time $t = t_{k-1}$, current state estimate $\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}(t_k^-)$ and error covariance matrix $\mathbf{P}_k^- = \mathbf{P}(t_k^-)$ can be obtained by integrating forward from $t = t_{k-1}$ to $t = t_k^-$ ($k = 1, 2, \cdots$) using the following state estimate propagation equation and error covariance propagation equation:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t), \tag{2.33}$$

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\, \mathbf{P}(t) + \mathbf{P}(t)\, \mathbf{F}^T(\hat{\mathbf{x}}(t), t) + \mathbf{G}(\mathbf{x}(t), t)\, \mathbf{Q}(t)\, \mathbf{G}^T(\mathbf{x}(t), t). \tag{2.34}$$

B. Measurement Update (Correction Step)

Given the time-updated state $\hat{\mathbf{x}}_k^-$ and the error covariance matrix $\mathbf{P}_k^-$ with new measurement vector $\mathbf{y}_k = \mathbf{y}(t_k)$, the measurement update step is performed in order to obtain the measurement-updated state and the error covariance matrix $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}(t_k), \mathbf{P}_k = \mathbf{P}(t_k)$, respectively. Using the available measurement $\mathbf{y}_k = \mathbf{y}(t_k)$ at time instant $t = t_k$, we can obtain the measurement-updated state estimate $\hat{\mathbf{x}}_k$ and the error covariance matrix $\mathbf{P}_k$ by the following measurement processing.

For the measurement update at time $t = t_k$, we can use

$$\mathbf{K}_k = \mathbf{P}_k^-\, \mathbf{H}_k^T(\hat{\mathbf{x}}_k^-) \left[\mathbf{H}_k(\hat{\mathbf{x}}_k^-)\, \mathbf{P}_k^-\, {\mathbf{H}_k}^T(\hat{\mathbf{x}}_k^-) + \mathbf{R}_k\right]^{-1}, \tag{2.35}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\, \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) \,\right], \tag{2.36}$$

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\, \mathbf{P}_k^- \, [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]^T + \mathbf{K}_k\mathbf{R}_k{\mathbf{K}_k}^T, \tag{2.37}$$

$$(\ \mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\, \mathbf{P}_k^-\ ). \tag{2.38}$$

Whenever a measurement is available at any time instant $t = t_k$, this measurement can be included in this measurement update processing. Since the dimension of the measurement vector varies depending on the availability of each sensor measurement at every moment, the dimensions of vectors $\mathbf{y}_k,\ \mathbf{h}_k$ and matrices $\mathbf{H}_k,\ \mathbf{R}_k,\ \mathbf{K}_k$ also vary.

2.3.2 Extended Kalman Filter with Sequential Measurement Updates

Let's consider the case where we have several sources of measurement inputs from different sensors. In general, each measurement may have different data size with different update rate. In order to effectively deal with this complex sensor fusion problem, we introduce the EKF with sequential measurement updates. The original approach for this filter design was to decompose the measurement vector into a series of scalar quantities and then to process these scalar measurements sequentially [2,13,111]. The main purpose of this scalar conversion was to reduce computational time and thus increase real-time capability by avoiding matrix inversion involved in the computation of a Kalman gain matrix. This filter can be extended to deal with measurements sequentially in smaller size measurement vectors and matrices instead of scalar quantities [7]. If we assign each sensor measurement to a sequential measurement update, we can effectively handle the sensor fusion problem with different data sizes and update rates.

The general nonlinear continuous-time process model and discrete-time measurement model in state-space form are given by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(\mathbf{x}(t), t)\ \mathbf{w}(t), \qquad \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t)), \tag{2.39}$$

$$\mathbf{y}_k^l = \mathbf{h}^l(\mathbf{x}_k, k) + \mathbf{v}_k^l, \qquad \mathbf{v}_k^l \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^l), \quad l = 1, 2, \cdots, r, \tag{2.40}$$

where r is the number of aiding sensor measurements. The initial conditions ($\mathbf{x}(t_0) \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$) are assumed to be known and have the following form:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}(t_0)], \quad \mathbf{P}_0 = E[(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)^T]. \tag{2.41}$$

Jacobian matrices of system dynamics and measurement model are defined as

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t)}, \quad \mathbf{H}_k^l = \frac{\partial \mathbf{h}^l(\mathbf{x}, k)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}(t_k^-)}, \quad l = 1, 2, \cdots, r. \tag{2.42}$$

A. Time Update (Prediction Step)

Given the state estimate $\hat{\mathbf{x}}_{k-1} = \hat{\mathbf{x}}(t_{k-1})$ and the error covariance matrix $\mathbf{P}_{k-1} = \mathbf{P}(t_{k-1})$ at time $t = t_{k-1}$, current state estimate $\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}(t_k^-)$ and error covariance matrix $\mathbf{P}_k^- = \mathbf{P}(t_k^-)$ can be obtained by integrating forward from $t = t_{k-1}$ to $t = t_k^-$ ($k = 1, 2, \cdots$) using the following state estimate propagation equation and error covariance propagation equation:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t), \tag{2.43}$$

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\, \mathbf{P}(t) + \mathbf{P}(t)\, \mathbf{F}^T(\hat{\mathbf{x}}(t), t) + \mathbf{G}(\mathbf{x}(t), t)\, \mathbf{Q}(t)\, \mathbf{G}^T(\mathbf{x}(t), t). \tag{2.44}$$

B. Sequential Measurement Update (Correction Step)

Given the time-updated state $\hat{\mathbf{x}}_k^-$ and the error covariance matrix $\mathbf{P}_k^-$ with new measurement vector $\mathbf{y}_k = \mathbf{y}(t_k) = \left[\mathbf{y}^1(t_k)^T \cdots \mathbf{y}^r(t_k)^T\right]^T$, we apply the ''sequential processing of measurement updates" in order to obtain the measurement-updated state and error covariance matrix $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}(t_k), \mathbf{P}_k = \mathbf{P}(t_k)$. For each available measurement $\mathbf{y}_k^l = \mathbf{y}^l(t_k)$ $(l = 1, \cdots, r)$ at time instant $t = t_k$, we can update the state estimate $\hat{\mathbf{x}}_k$ and the error covariance matrix $\mathbf{P}_k$ by the following sequential measurement processing:

For $l = 1, 2, \cdots, r$ (r measurements update at time $t = t_k$),

$$\mathbf{K}_k^l = \mathbf{P}_k^{l-1}\, \mathbf{H}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1}) \left[\mathbf{H}_k^l(\hat{\mathbf{x}}_k^{l-1})\, \mathbf{P}_k^{l-1}\, \mathbf{H}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1}) + \mathbf{R}_k^l\right]^{-1}, \tag{2.45}$$

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l \left[\, \mathbf{y}_k^l - \mathbf{h}^l(\hat{\mathbf{x}}_k^{l-1}) \,\right], \tag{2.46}$$

$$\mathbf{P}_k^l = \left[\mathbf{I} - \mathbf{K}_k^l \mathbf{H}_k^l\right] \mathbf{P}_k^{l-1} \left[\mathbf{I} - \mathbf{K}_k^l \mathbf{H}_k^l\right]^T + \mathbf{K}_k^l \mathbf{R}_k^l \mathbf{K}_k^{l\,T}, \tag{2.47}$$

$$\left(\mathbf{P}_k^l = \left[\mathbf{I} - \mathbf{K}_k^l \mathbf{H}_k^l\right] \mathbf{P}_k^{l-1} \right), \quad l = 1, 2, \cdots, r, \tag{2.48}$$

where the starting initial condition for this sequential measurement update at $t = t_k^-$ is $\hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}(t_k^-)$, $\mathbf{P}_k^0 = \mathbf{P}_k^- = \mathbf{P}(t_k^-)$ and the final measurement update is set to $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}(t_k) = \hat{\mathbf{x}}_k^r$, $\mathbf{P}_k = \mathbf{P}(t_k) = \mathbf{P}_k^r$.

Whenever a measurement is available at any time instant $t = t_k$, this measurement can be included in this sequential measurement update processing. For every

measurement not available at time $t = t_k$, we can skip this measurement update step. The dimensions of all measurement-related vectors and matrices are fixed.

For more detailed derivation and theoretical explanation, refer to the Appendix A.

2.4 Unscented Kalman Filters

This section presents the standard unscented Kalman filter (UKF) and a newly-developed UKF with sequential measurement updates. The UKF is known to provide accurate estimation results comparable to at least second-order filters, while the EKF provides only first-order accuracy. Moreover, the UKF is easier to implement than the EKF since it does not require the computation of Jacobian matrices whose analytical expression is often very difficult to derive. After introducing the algorithm of the standard UKF in the beginning of the section, a new UKF with sequential measurement updates is developed to deal with the complicated multi-sensor fusion problems mentioned in the previous section. This filter can effectively handle the multi-sensor fusion of different measurement size and data rate while keeping the advantages of the standard UKF such as accurate estimation and no Jacobian matrix computations.

2.4.1 Unscented Transformation

Since the unscented transformation is the basic concept for the development of the UKF, this section presents some details of it. The unscented transformation (UT) is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation [39]. Consider a nonlinear function

$$\mathbf{y} = \mathbf{f}(\mathbf{x}), \tag{2.49}$$

where $\mathbf{x}$ is a random variable (dimension, n). We assume $\mathbf{x}$ has mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}}$. Now, what are the statistics of the random variable $\mathbf{y}$? Or what are the mean $\bar{\mathbf{y}}$

and the covariance $\mathbf{P_y}$ of $\mathbf{y}$? For this purpose we first form a sigma point matrix $\mathbf{X}$ composed of $(2n+1)$ sigma point vectors $\mathbf{X}_i$ as shown below:

$$\mathbf{X}_0 = \bar{\mathbf{x}}, \tag{2.50}$$

$$\mathbf{X}_i = \bar{\mathbf{x}} + \gamma \left(\sqrt{\mathbf{P_x}}\right)_i, \; i = 1, \cdots, n, \tag{2.51}$$

$$\mathbf{X}_i = \bar{\mathbf{x}} - \gamma \left(\sqrt{\mathbf{P_x}}\right)_{i-n}, \; i = n+1, \cdots, 2n, \tag{2.52}$$

where the constant $\gamma = \sqrt{n+\lambda}$ determines the spread of sigma points around mean state value $\bar{\mathbf{x}}$, and $\left(\sqrt{\mathbf{P_x}}\right)_i$ is the ith column of the matrix square root (e.g., lower-triangular Cholesky factorization) of the covariance matrix $\mathbf{P_x} = \left(\sqrt{\mathbf{P_x}}\right)\left(\sqrt{\mathbf{P_x}}\right)^T$. λ is selected to better approximate the higher-order moments of the nonlinear transformation which depend on statistical characteristics or specific probability distribution. The sigma point vectors are then mapped through the nonlinear function

$$\mathbf{Y}_i = \mathbf{f}(\mathbf{X}_i), \; i = 0, \cdots, 2n. \tag{2.53}$$

Finally, the mean and covariance of $\mathbf{y}$ are approximated using a weighted sample mean and covariance of the posterior sigma points as follows.

$$\bar{\mathbf{y}} \simeq \sum_{i=0}^{2n} w_i^{(m)} \mathbf{Y}_i, \tag{2.54}$$

$$\mathbf{P_y} \simeq \sum_{i=0}^{2n} w_i^{(c)} (\mathbf{Y}_i - \bar{\mathbf{y}}) \, (\mathbf{Y}_i - \bar{\mathbf{y}})^T, \tag{2.55}$$

where $w_i^{(m)}$ and $w_i^{(c)}$ are constant weights defined later.

This relatively simple approach is known to have third-order approximation accuracy for Gaussian inputs for all nonlinearities, and at least second-order accuracy for non-Gaussian inputs [59].

A. Choice of Parameters and Weights in the Original Unscented Transformation

In order to apply the unscented transformation to the UKF, it is necessary to select several parameters and weights [59]. When the random variable $\mathbf{x}$ is assumed to be

Gaussian, a general choice for the constant γ, needed to determine the spread of sigma points around mean state value, is determined so that $n + \lambda = 3$, and thus $\gamma = \sqrt{n + \lambda} = \sqrt{3}$. In addition, following weights are used:

$$w_0^{(m)} = \frac{\lambda}{n + \lambda} = 1 - \frac{n}{\gamma^2}, \tag{2.56}$$

$$w_0^{(c)} = \frac{\lambda}{n + \lambda} = 1 - \frac{n}{\gamma^2}, \tag{2.57}$$

$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(n + \lambda)} = \frac{1}{2\gamma^2}, \quad i = 1, 2, \cdots, 2n. \tag{2.58}$$

B. Choice of Parameters and Weights in the Scaled Unscented Transformation

When the dimension (n) of the involved random variable $\mathbf{x}$ is greater than 3 (i.e., $n > 3$), then λ becomes negative for the specific case of $n + \lambda = 3$ ($\lambda = 3 - n < 0$) where $w_0^{(c)}$ are negative. With this choice of weight, the computed covariance $\mathbf{P_y}$ can be non-positive semi-definite [115]. In order to overcome this problem, the scaled unscented transformation is developed where weights have the following forms [50]:

$$w_0^{(m)} = \frac{\lambda}{n + \lambda} = 1 - \frac{n}{\gamma^2}, \tag{2.59}$$

$$w_0^{(c)} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta) = w_0^{(m)} + (3 - \alpha^2), \tag{2.60}$$

$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(n + \lambda)} = \frac{1}{2\gamma^2}, \quad i = 1, 2, \cdots, 2n. \tag{2.61}$$

Here, $\lambda = \alpha^2(n + \kappa) - n$ is a scaling parameter and α determines the spread of the sigma points around $\bar{\mathbf{x}}$ and usually set to a small positive value, $10^{-4} \leq \alpha \leq 1$. The constant κ is a secondary scaling parameter, usually set to zero. β is chosen to be 2 ($\beta = 2$ is optimal for Gaussian distribution) in order to incorporate prior knowledge of the distribution of $\mathbf{x}$. Considering these choices of parameters, the constant γ, needed to determine the spread of sigma points around mean state value, becomes $\gamma(= \sqrt{n + \lambda}) = \alpha \sqrt{n}$ where α is the only parameter to control the spread of the sigma points around mean ($\alpha = 0.001$ is used for our simulations.).

2.4.2 Accuracy Analysis of Unscented Transformation

This section presents some discussions on the accuracy analysis of the unscented transformation by comparing it with the Monte Carlo method and the linearization method. The Monte Carlo method is known to be most accurate for calculating the statistical characteristics of a nonlinear function of a random variable if enough independent and identically distributed (i.i.d.) random samples are chosen. According to the strong law of large numbers, statistical estimate given by the Monte Carlo method converges to the true expectation almost surely as the number of i.i.d. random samples goes to infinity [23]. As an illustration of computing the statistical characteristics in a nonlinear function of a random variable, the mean and covariance of the coordinate transformation from the polar frame to the Cartesian frame is computed by the linearization method (used in the EKF), the unscented transformation (used in the UKF), and the Monte Carlo (MC) method. This example is closely related to current work since many sensor measurements in target tracking problems are given in polar coordinates (e.g., range and bearings) and estimates are provided in Cartesian coordinates (e.g., position and velocity). In fact, our vision-based tracking system which will be discussed in detail in Chapters 4 and 5 falls into the same category since sensor measurements are angles such as azimuth, elevation, and subtended angles and Cartesian position and velocity are estimated based on the angle measurements.

Let's consider the following stochastic nonlinear transformation that represents the coordinate transformation from polar to Cartesian frame.

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \tag{2.62}$$

or

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} x_1 \cos x_2 \\ x_1 \sin x_2 \end{bmatrix}, \tag{2.63}$$

where $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r \\ \theta \end{bmatrix}$ is a point in the two-dimensional polar coordinate system, and $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r\,\cos\theta \\ r\,\sin\theta \end{bmatrix}$ is a point in the Cartesian coordinate system.

■ *Problem*: Given the statistics of the random variable $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P_x})$, we need to compute the statistics of $\mathbf{y}$ (nonlinear function of $\mathbf{x}$, $\mathbf{y} = \mathbf{f}(\mathbf{x})$) where $\mathbf{y} \sim \mathcal{N}(\bar{\mathbf{y}}, \mathbf{P_y})$.

Now, for given Gaussian distribution $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P_x})$, means and covariances of the nonlinear transformation are computed by the three approaches described below.

A. Monte Carlo (MC) method

For each sample point $\mathbf{x}^i,\ i = 1, 2, \cdots N$ (N: the number of sample points), drawn from the Gaussian distribution $\mathcal{N}(\bar{\mathbf{x}}, \mathbf{P_x})$, $\mathbf{y}^i$ are computed based on the nonlinear function:

$$\mathbf{y}^i = \mathbf{f}(\mathbf{x}^i),\ i = 1, 2, \cdots N, \tag{2.64}$$

where $\mathbf{x}^i \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P_x})$. Then, MC mean and MC covariance are computed by

$$\bar{\mathbf{y}}_{MC} = \frac{1}{N} \sum_{i=0}^{N} \mathbf{y}^i, \tag{2.65}$$

$$\mathbf{P}_{\mathbf{y}MC} = \frac{1}{N} \sum_{i=0}^{N} [\mathbf{y}^i - \bar{\mathbf{y}}_{MC}]\,[\mathbf{y}^i - \bar{\mathbf{y}}_{MC}]^T. \tag{2.66}$$

The appropriateness of selected samples can be evaluated by comparing the MC mean $\bar{\mathbf{x}}_{MC}$ and MC covariance $\mathbf{P}_{\mathbf{x}MC}$ of the random variable $\mathbf{x}$ with the original mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P_x}$. The MC mean and MC covariance of the samples $\mathbf{x}^i$ can be computed

by

$$\bar{\mathbf{x}}_{MC} = \frac{1}{N}\sum_{i=0}^{N}\mathbf{x}^i, \tag{2.67}$$

$$\mathbf{P}_{\mathbf{x}MC} = \frac{1}{N}\sum_{i=0}^{N}[\mathbf{x}^i - \bar{\mathbf{x}}_{MC}]\,[\mathbf{x}^i - \bar{\mathbf{x}}_{MC}]^T. \tag{2.68}$$

B. Linearization method (EKF method)

The mean and covariance in the linearization method are the first-order terms in the Taylor series expansions as follows [39].

$$\bar{\mathbf{y}}_{Lin} = \mathbf{f}(\bar{\mathbf{x}}), \tag{2.69}$$

$$\mathbf{P}_{\mathbf{y}Lin} = \mathbf{F}\,\mathbf{P}_{\mathbf{x}}\,\mathbf{F}^T, \tag{2.70}$$

where $\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\mathbf{x}=\bar{\mathbf{x}}} = \begin{bmatrix} \cos x_2 & -x_1\,\sin x_2 \\ \sin x_2 & x_1\,\cos x_2 \end{bmatrix}_{\mathbf{x}=\bar{\mathbf{x}}}$.

C. Unscented transformation method (UKF method)

$$\mathbf{P}_{\mathbf{x}} = \mathbf{S}\,\mathbf{S}^T = \left(\sqrt{\mathbf{P}_{\mathbf{x}}}\right)\left(\sqrt{\mathbf{P}_{\mathbf{x}}}\right)^T, \tag{2.71}$$

$$\mathbf{X} = [\ \bar{\mathbf{x}},\ \bar{\mathbf{x}} + \gamma\,\mathbf{S},\ \bar{\mathbf{x}} - \gamma\,\mathbf{S}\ ], \tag{2.72}$$

$$\mathbf{Y}_i = \mathbf{f}(\mathbf{X}_i),\ i = 0, 1, 2, \cdots 2n, \tag{2.73}$$

$$\bar{\mathbf{y}}_{UT} = \sum_{i=0}^{2n} w_i^{(m)}\ \mathbf{Y}_i, \tag{2.74}$$

$$\mathbf{P}_{\mathbf{y}UT} = \sum_{i=0}^{2n} w_i^{(c)}\ [\ \mathbf{Y}_i - \bar{\mathbf{y}}_{UT}\ ]\,[\ \mathbf{Y}_i - \bar{\mathbf{y}}_{UT}\ ]^T. \tag{2.75}$$

D. Numerical results

For given Gaussian variable $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P}_{\mathbf{x}})$ with

$$\bar{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{\pi}{2} \end{bmatrix} \simeq \begin{bmatrix} 1 \\ 1.57079632679490 \end{bmatrix}, \tag{2.76}$$

$$\mathbf{P}_{\mathbf{x}} = \begin{bmatrix} 0.02^2 & 0.002 \\ 0.002 & (10\ \text{deg})^2 \end{bmatrix} \simeq \begin{bmatrix} 0.0004 & 0.002 \\ 0.002 & 0.03046174197867 \end{bmatrix}, \tag{2.77}$$

computed means and covariances are as follows:

$$\begin{aligned}
\bar{\mathbf{x}}_{MC} &= \begin{bmatrix} 0.99977136815831 \\ 1.56822473326594 \end{bmatrix}, \quad \mathbf{P}_{\mathbf{x}MC} = \begin{bmatrix} 0.00039849370171 & 0.00196067179446 \\ 0.00196067179446 & 0.03107659673404 \end{bmatrix}, \\
\bar{\mathbf{y}}_{MC} &= \begin{bmatrix} 0.00052458550136 \\ 0.98436469563870 \end{bmatrix}, \quad \mathbf{P}_{\mathbf{y}MC} = \begin{bmatrix} 0.03012625458037 & -0.00207039730555 \\ -0.00207039730555 & 0.00084089850067 \end{bmatrix}, \\
\bar{\mathbf{y}}_{Lin} &= \begin{bmatrix} 0.00000000000000 \\ 1.00000000000000 \end{bmatrix}, \quad \mathbf{P}_{\mathbf{y}Lin} = \begin{bmatrix} 0.03046174197867 & -0.00200000000000 \\ -0.00200000000000 & 0.00040000000000 \end{bmatrix}, \\
\bar{\mathbf{y}}_{UT} &= \begin{bmatrix} -0.00199001498929 \\ 0.98483384490800 \end{bmatrix}, \quad \mathbf{P}_{\mathbf{y}UT} = \begin{bmatrix} 0.02995479036525 & -0.00196064451753 \\ -0.00196064451753 & 0.00054354739882 \end{bmatrix}.
\end{aligned} \tag{2.78}$$

Figure 2.2 shows the numerical results in polar coordinates. Green dots are i.i.d. random samples $\mathbf{x}^i$ ($N = 2000$) based on true mean $\bar{\mathbf{x}}$ (red circle which coincides with a blue diamond) and true covariance $\mathbf{P}_{\mathbf{x}}$ (2σ shown in red ellipse, $(\mathbf{x}-\bar{\mathbf{x}})^T \mathbf{P}_{\mathbf{x}}{}^{-1} (\mathbf{x}-\bar{\mathbf{x}}) = 2^2$). Using the i.i.d. random samples, the MC mean $\bar{\mathbf{x}}_{MC}$ (blue diamond) and MC covariance $\mathbf{P}_{\mathbf{x}MC}$ (2σ ellipse in blue dashed line) are computed and presented in the same figure. Mean and covariance of the Monte Carlo method agree well with those of true values, which shows the chosen random samples properly describe the statistical characteristics of the problem. Five 'x' symbols in pink color (one coincides with blue diamond) represent the sigma points $\mathbf{X}_i$ used in the UT method.

Figure 2.3 compares the numerical results in the Cartesian coordinates obtained through the polar-to-Cartesian nonlinear transformation. Green dots are transformed

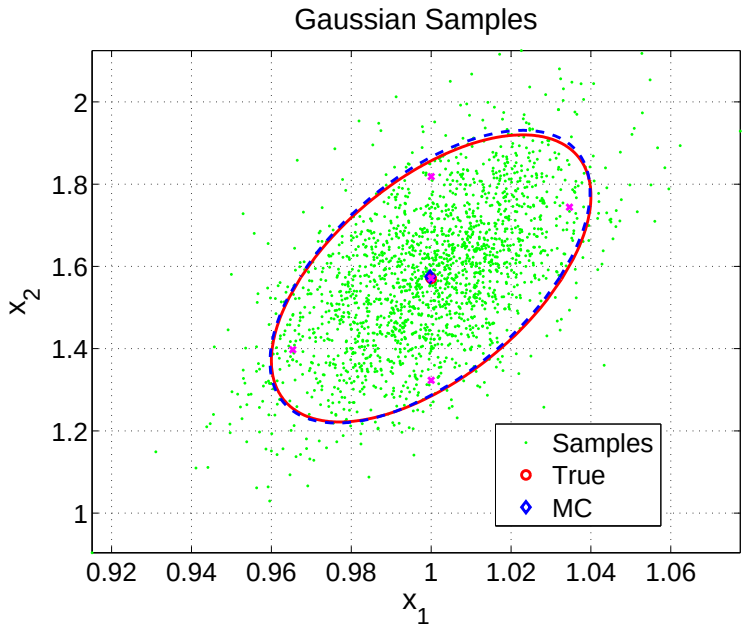


Figure 2.2: (r, θ) in the Polar Coordinates

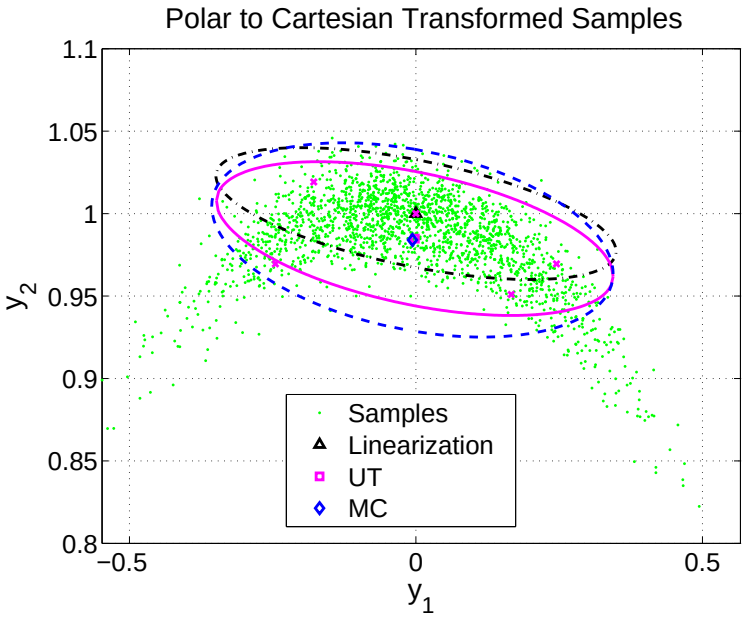


Figure 2.3: $(x, y) = (r\cos\theta, r\sin\theta)$ in the Cartesian Coordinates

samples $\mathbf{y}^i$. Blue diamond and blue dashed line are the transformed MC mean $\bar{\mathbf{y}}_{MC}$ and 2σ MC covariance $\mathbf{P}_{\mathbf{y}_{MC}}$, respectively, computed by the transformed samples. The transformed UT mean $\bar{\mathbf{y}}_{UT}$ (pink square hidden behind a blue diamond) agrees well with that of transformed MC mean $\bar{\mathbf{y}}_{MC}$ (blue diamond), while the transformed mean $\bar{\mathbf{y}}_{Lin}$ (black triangle) computed by the linearization method deviates a lot from that of the Monte Carlo method. The 2σ covariance of the UT method (pink ellipse of $(\mathbf{y}-\bar{\mathbf{y}}_{UT})^T \mathbf{P}_{\mathbf{y}_{UT}}^{-1} (\mathbf{y}-\bar{\mathbf{y}}_{UT}) = 2^2$) better represents the 2σ covariance of the Monte Carlo method (blue dashed ellipse of $(\mathbf{y}-\bar{\mathbf{y}}_{MC})^T \mathbf{P}_{\mathbf{y}_{MC}}^{-1} (\mathbf{y}-\bar{\mathbf{y}}_{MC}) = 2^2$) compared to that of the linearization method (black ellipse of $(\mathbf{y}-\bar{\mathbf{y}}_{Lin})^T \mathbf{P}_{\mathbf{y}_{Lin}}^{-1} (\mathbf{y}-\bar{\mathbf{y}}_{Lin}) = 2^2$). Five 'x' pink symbols (one coincides with black triangle) represent the transformed sigma points $\mathbf{Y}_i$ used to approximate the statistics of the nonlinear transformation by the UT method.

From this practical example of the polar-to-Cartesian nonlinear transformation in Figure 2.3, the UT method provides better accuracy than the linearization method in estimating the statistical characteristics of a stochastic nonlinear transformation. Since the UKF is based on the UT method and the EKF on the linearization method, it is natural to suppose that the UKF will have better estimation performance than the EKF in nonlinear filtering problems.

From Eq. 2.69 and Figure 2.3, we see the fact that the nonlinear transformed point $\mathbf{f}(\bar{\mathbf{x}})$ of a true mean in the input plane is far from the true mean $\bar{\mathbf{y}}$ in the transformed plane (i.e., $\bar{\mathbf{y}} \neq \mathbf{f}(\bar{\mathbf{x}})$, where $\bar{\mathbf{y}}_{Lin} \triangleq \mathbf{f}(\bar{\mathbf{x}})$ and $\bar{\mathbf{y}} \cong \bar{\mathbf{y}}_{MC}$). In the transformed plane, this point $\mathbf{f}(\bar{\mathbf{x}})$ is just one of many transformed points, which does not provide complete statistics in the transformed plane. Covariance effect is not included in the computation of this point.

In the case of the UT, a mean and several (deterministically) perturbed points around the mean are first selected and these points undergo the nonlinear transformation. Statistics in the transformed plane are calculated based on transformed points.

Covariance is used in selecting perturbed points (or sigma points). Since both mean and covariance are used in the nonlinear transformation, the UT better conserves statistics in the transformed plane. Depending on the nonlinearity of the nonlinear function, distribution of transformed sample points is distorted in the transformed plane. Since the UT includes this nonlinear distribution effect in the transformed plane through covariance, it reduces bias from the true mean in the transformed plane.

Let's expand the true mean $\bar{\mathbf{y}}$ in the transformed plane as a function of statistics (mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}}$) in the input plane.

$$\bar{\mathbf{y}} = \mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{f}(\mathbf{x})] = \mathbb{E}[\mathbf{f}(\bar{\mathbf{x}} + \delta\mathbf{x})] \tag{2.79}$$

$$= \mathbb{E}[\ \mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{1!}\left[(\delta\mathbf{x} \cdot \nabla_{\mathbf{x}})\mathbf{f}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} + \frac{1}{2!}\left[(\delta\mathbf{x} \cdot \nabla_{\mathbf{x}})^2\mathbf{f}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} + \cdots\ ] \tag{2.80}$$

$$= \mathbf{f}(\bar{\mathbf{x}}) + \mathbb{E}[\ \left[(\delta\mathbf{x} \cdot \nabla_{\mathbf{x}})\mathbf{f}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} + \frac{1}{2!}\left[(\delta\mathbf{x} \cdot \nabla_{\mathbf{x}})^2\mathbf{f}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} + \cdots\ ] \tag{2.81}$$

$$= \mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{2}\left[(\nabla_{\mathbf{x}}^T\mathbf{P}_{\mathbf{x}}\nabla_{\mathbf{x}})\mathbf{f}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} + \mathbb{E}[\ \frac{1}{4!}\left[(\delta\mathbf{x} \cdot \nabla_{\mathbf{x}})^4\mathbf{f}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} + \cdots\ ], \tag{2.82}$$

where $\mathbf{x}$ is a random variable with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}} = \mathbb{E}[\delta\mathbf{x}\ \delta\mathbf{x}^T]$. All odd moments are zero since we assume $\mathbf{x}$ is a symmetrically distributed random variable (e.g., Gaussian distribution).

In Eq. 2.82, we can observe, in addition to $\mathbf{f}(\bar{\mathbf{x}})$, the second-order effect of covariance in computing the true mean in the transformed plane. For more general mathematical analysis regarding the accuracy of the unscented transformation and the scaled unscented transformation, refer to the references [50, 52, 115, 124].

2.4.3 Standard Unscented Kalman Filter (UKF)

This section summarizes the standard UKF, first developed by Julier et al. [53, 58, 59] and diversified into several algorithms by Merwe et al. [116, 118, 119]. This filter is applied to the design of a new vision-based tracking system in Chapter 4. This filter is also used as part of a new extended marginalized particle filter in order to deal

with nonlinear internal process dynamics with Gaussian process noise.

The general nonlinear discrete-time process model and discrete-time measurement model in state-space form are given by

$$\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \tag{2.83}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{2.84}$$

where initial conditions ($\mathbf{x}(t_0) \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$) are assumed to be known.

In order to start the standard UKF algorithm, we need to initialize $\hat{\mathbf{x}}_0$ and $\mathbf{S}_0$ as follows:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}(t_0)], \tag{2.85}$$

$$\mathbf{P}_0 = E[(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)^T], \tag{2.86}$$

$$\mathbf{S}_0 = \{chol(\mathbf{P}_0)\}^T. \tag{2.87}$$

Note that Matlab function "chol" provides Cholesky factorization in upper triangular form, but we need it transposed to obtain $\mathbf{S}_0$ in lower triangular form.

A. Time Update (Prediction Step)

For each time step $k = 1, 2, \cdots$, we need first to calculate sigma points and then time-propagate using time-update equations:

• **Sigma-point calculation**

$$\mathbf{S}_{k-1} = \{chol(\mathbf{P}_{k-1})\}^T, \tag{2.88}$$

$$\mathbf{X}_{k-1} = [\ \hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} + \gamma\, \mathbf{S}_{k-1} \quad \hat{\mathbf{x}}_{k-1} - \gamma\, \mathbf{S}_{k-1}\ ]. \tag{2.89}$$

• **Time-update equations**

$$\mathbf{X}^*_{k|k-1} = \mathbf{f}_d(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}), \tag{2.90}$$

$$\hat{\mathbf{x}}^-_k = \sum_{i=0}^{2n} w_i^{(m)} \mathbf{X}^*_{i,k|k-1}, \tag{2.91}$$

$$\mathbf{P}^-_k = \sum_{i=0}^{2n} w_i^{(c)} (\mathbf{X}^*_{i,k|k-1} - \hat{\mathbf{x}}^-_k)(\mathbf{X}^*_{i,k|k-1} - \hat{\mathbf{x}}^-_k)^T + \mathbf{Q}_k, \tag{2.92}$$

where $\mathbf{Q}_k = \Delta t \; \mathbf{Q}(t)$.

B. Measurement Update (Correction Step)

• **Augmented sigma points**[3]

$$\mathbf{S}^-_k = \{chol(\mathbf{P}^-_k)\}^T, \tag{2.93}$$

$$\mathbf{X}_{k|k-1} = [\; \hat{\mathbf{x}}^-_k \quad \hat{\mathbf{x}}^-_k + \gamma \, \mathbf{S}^-_k \quad \hat{\mathbf{x}}^-_k - \gamma \, \mathbf{S}^-_k \;], \tag{2.94}$$

$$\mathbf{Y}_{k|k-1} = \mathbf{h}(\mathbf{X}_{k|k-1}), \tag{2.95}$$

$$\hat{\mathbf{y}}^-_k = \sum_{i=0}^{2n} w_i^{(m)} \mathbf{Y}_{i,k|k-1}. \tag{2.96}$$

• **Measurement-update equations**

$$\mathbf{P}_y = \sum_{i=0}^{2n} w_i^{(c)} (\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}^-_k)(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}^-_k)^T + \mathbf{R}_k, \tag{2.97}$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} w_i^{(c)} (\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}^-_k)(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}^-_k)^T, \tag{2.98}$$

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_y^{-1}, \tag{2.99}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^-_k + \mathbf{K}_k \left(\, \mathbf{y}_k - \hat{\mathbf{y}}^-_k \, \right), \tag{2.100}$$

$$\mathbf{P}_k = \mathbf{P}^-_k - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T. \tag{2.101}$$

[3] One choice is $\mathbf{X}_{k|k-1} = \mathbf{X}_{k-1}$ [56]. Merwe et al. [124] suggests $\mathbf{X}_{k|k-1} = [\; \mathbf{X}^*_{k|k-1} \quad \mathbf{X}^*_{k|k-1} + \gamma \sqrt{\mathbf{Q}_k} \quad \mathbf{X}^*_{k|k-1} - \gamma \sqrt{\mathbf{Q}_k} \;]$, but this method is computationally expensive. $\mathbf{X}_{k|k-1} = [\; \hat{\mathbf{x}}^-_k \quad \hat{\mathbf{x}}^-_k + \gamma \sqrt{\mathbf{Q}_k} \quad \hat{\mathbf{x}}^-_k - \gamma \sqrt{\mathbf{Q}_k} \;]$ may also be a selection.

2.4.4 Unscented Kalman Filter with Sequential Measurement Update

This section introduces the newly-developed UKF with sequential measurement updates. The idea behind this is to recursively update the state estimate and error covariance matrix by using more accurate nonlinear filtering approach, the UKF, while maintaining the advantages of sequential measurement updates mentioned in the Section 2.3.2. By using a new UKF with sequential measurement updates instead of the standard UKF algorithm, we can keep the advantages of the EKF with sequential measurement updates. In this framework, we can easily handle the multi-rate sensor fusion problem and sensor latency compensation compared to that of the standard UKF. The addition of new aiding sensors is easier in this framework than in the standard UKF framework. Furthermore, we can remove the computational step of messy Jacobian matrices and keep at least second-order nonlinear function approximation.

The general nonlinear discrete-time process model and discrete-time measurement model in state-space form are given by

$$\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \tag{2.102}$$

$$\mathbf{y}_k^l = \mathbf{h}^l(\mathbf{x}_k, k) + \mathbf{v}_k^l, \qquad \mathbf{v}_k^l \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^l), \;\; l = 1, 2, \cdots, r, \tag{2.103}$$

where initial conditions ($\mathbf{x}(t_0) \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$) are assumed to be known and r is the number of aiding sensor measurements.

A. Time Update (Prediction Step)

For each time step $k = 1, 2, \cdots$, we need first to calculate sigma points and then time-update using time-update equations.

• Sigma-point calculation

$$\mathbf{S}_{k-1} = \{chol(\mathbf{P}_{k-1})\}^T, \tag{2.104}$$

$$\mathbf{X}_{k-1} = [\, \hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} + \gamma\, \mathbf{S}_{k-1} \quad \hat{\mathbf{x}}_{k-1} - \gamma\, \mathbf{S}_{k-1} \,], \tag{2.105}$$

where $\mathbf{S}$ is Cholesky factorization of $\mathbf{P}$ in lower triangular matrix form and the constant γ is a parameter to control the dispersion distance from mean estimate in the computation of sigma point matrix $\mathbf{X}$.

• **Time-update equations**[4]

$$\mathbf{X}^*_{k|k-1} = \mathbf{f}_d(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}) \;\text{ or }\; \mathbf{X}^*_{k|k-1} = \mathbf{X}_{k-1} + \int_{t_{k-1}}^{t_k^-} \mathbf{f}(\mathbf{X}_{k-1}, t)\, dt \quad \text{with Modified Euler,} \tag{2.106}$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} w_i^{(m)} \mathbf{X}^*_{i,k|k-1}, \tag{2.107}$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2n} w_i^{(c)} (\mathbf{X}^*_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\mathbf{X}^*_{i,k|k-1} - \hat{\mathbf{x}}_k^-)^T + \mathbf{Q}_k, \tag{2.108}$$

where $\mathbf{Q}_k = \Delta t\ \mathbf{Q}(t)$.

B. Sequential Measurement Update at time instant ($t = t_k$)

For each available measurement $\mathbf{y}_k^l = \mathbf{y}^l(t_k)$ $(l = 1, \cdots, r)$ at time instant $t = t_k$, we can update the state estimate $\hat{\mathbf{x}}_k$ and error covariance matrix $\mathbf{P}_k$ by the following sequential measurement processing. We choose the time update quantity as the initial conditions for the measurement update and then process the available measurements sequentially or one by one.

Let the initial conditions be $\hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_k^-$, $\mathbf{P}_k^0 = \mathbf{P}_k^-$ and iterate the measurement update loop for $l = 1, 2, \cdots, r$ (r measurement updates at time $t = t_k$).

• **Augmented sigma points**

$$\mathbf{S}_k^{l-1} = \sqrt{\mathbf{P}_k^{l-1}} = \{chol(\mathbf{P}_k^{l-1})\}^T, \tag{2.109}$$

$$\mathbf{X}_k^{l-1} = [\ \hat{\mathbf{x}}_k^{l-1} \quad \hat{\mathbf{x}}_k^{l-1} + \gamma\, \mathbf{S}_k^{l-1} \quad \hat{\mathbf{x}}_k^{l-1} - \gamma\, \mathbf{S}_k^{l-1}\ ], \tag{2.110}$$

[4]In the design of an integration navigation system described in detail in Chapter 3, a modified Euler algorithm (or trapezoidal integration algorithm) is applied to the sigma point propagations instead of a Euler algorithm (or rectangular integration algorithm) in order to improve time propagation accuracy.

$$\mathbf{Y}_k^l = \mathbf{h}^l(\mathbf{X}_k^{l-1}), \tag{2.111}$$

$$\hat{\mathbf{y}}_k^l = \sum_{i=0}^{2n} w_i^{(m)} (\mathbf{Y}_k^l)_i. \tag{2.112}$$

• **Measurement-update equations**

$$\mathbf{P}_{y_k}^l = \sum_{i=0}^{2n} w_i^{(c)} \left[(\mathbf{Y}_k^l)_i - \hat{\mathbf{y}}_k^l \right] \left[(\mathbf{Y}_k^l)_i - \hat{\mathbf{y}}_k^l \right]^T + \mathbf{R}_k^l, \tag{2.113}$$

$$\mathbf{P}_{x_k y_k}^l = \sum_{i=0}^{2n} w_i^{(c)} \left[(\mathbf{X}_k^{l-1})_i - \hat{\mathbf{x}}_k^{l-1} \right] \left[(\mathbf{Y}_k^l)_i - \hat{\mathbf{y}}_k^l \right]^T, \tag{2.114}$$

$$\mathbf{K}_k^l = \mathbf{P}_{x_k y_k}^l (\mathbf{P}_{y_k}^l)^{-1}, \tag{2.115}$$

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l \left(\mathbf{y}_k^l - \hat{\mathbf{y}}_k^l \right), \tag{2.116}$$

$$\mathbf{P}_k^l = \mathbf{P}_k^{l-1} - \mathbf{K}_k^l \mathbf{P}_{y_k}^l {\mathbf{K}_k^l}^T. \tag{2.117}$$

After finishing the last measurement-update loop, the state and covariance estimates at time $t = t_k$ are given by $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^r = \hat{\mathbf{x}}^r(k)$, $\mathbf{P}_k = \mathbf{P}_k^r$. These results become the initial condition for the next time-update step.

For every measurement available at any time instant $t = t_k$, this measurement can be included in this sequential measurement update processing.

2.5 Sequential Monte Carlo Methods (Particle Filters)

In the general Bayesian estimation problem, the knowledge of a joint posterior pdf $p(\mathbf{X}_k|\mathbf{Y}_k)$ provides the complete solution to the calculation of any averaged statistics of the vector-valued function $\mathbf{g}(\mathbf{X}_k)$ of the state history $\mathbf{X}_k$ as follows:

$$\mathbb{I}[\mathbf{g}(\mathbf{X}_k)] \triangleq \hat{\mathbf{g}}(\mathbf{X}_k) \triangleq \mathbb{E}[\mathbf{g}(\mathbf{X}_k)|\mathbf{Y}_k] = \mathbb{E}_{p(\mathbf{X}_k|\mathbf{Y}_k)}[\mathbf{g}(\mathbf{X}_k)] = \int \mathbf{g}(\mathbf{X}_k)\, p(\mathbf{X}_k|\mathbf{Y}_k)\, d\mathbf{X}_k, \tag{2.118}$$

where $\mathbf{X}_k \triangleq \mathbf{x}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$ is the sequence of all available system states up to time k, and $\mathbf{Y}_k \triangleq \mathbf{y}_{1:k} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$ is the sequence of all available

measurements (or observations) up to time k. In the special case of a Bayesian filtering problem, the marginal posterior pdf (or filtering pdf) $p(\mathbf{x}_k|\mathbf{Y}_k)$ is enough to provide the complete solution for computing some averaged statistics of the vector-valued function $\mathbf{g}(\mathbf{x}_k)$ of the state $\mathbf{x}_k$ at time k as follows:

$$\mathbb{I}[\mathbf{g}(\mathbf{x}_k)] \triangleq \hat{\mathbf{g}}(\mathbf{x}_k) \triangleq \mathbb{E}[\mathbf{g}(\mathbf{x}_k)|\mathbf{Y}_k] = \mathbb{E}_{p(\mathbf{x}_k|\mathbf{Y}_k)}[\mathbf{g}(\mathbf{x}_k)] = \int \mathbf{g}(\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{Y}_k)\ d\mathbf{x}_k. \tag{2.119}$$

The sequential Monte Carlo methods recursively update the posterior density/distribution functions ($p(\mathbf{X}_k|\mathbf{Y}_k)$ or $p(\mathbf{x}_k|\mathbf{Y}_k)$) in the Bayesian estimation framework in which the posterior density functions are directly approximated by a set of weighted samples (or particles) based on Monte Carlo integration methods. Since these methods do not make any assumption on the form of the posterior density functions, they can be applied to a general class of nonlinear, non-Gaussian dynamic systems. The sequential Monte Carlo methods provide a powerful nonlinear estimation framework by combining the Monte Carlo sampling methods with Bayesian inference. The Monte Carlo sampling methods allow us to approximate complex-shaped posterior density functions without restrictions, and thus to handle the analytically intractable integrals of posterior density functions. Moreover, the Bayesian inference furnishes the recursive online estimation framework. Specifically, the Bayesian inference makes us solve this difficult nonlinear estimation problem recursively with a state transition model, an observation map, and current measurements without the need to save all the previous states and measurements. In order to comprehend the new type of particle filter described in the next section, we need some preliminary knowledge about the Monte Carlo integration method, recursive Bayesian inference, and the fundamental concepts of particle filtering [18, 19, 23, 33, 100, 115].

2.5.1 Perfect Monte Carlo Sampling

In order to estimate the averaged statistics (or expectation) of some vector-valued function of the state (history), we need to calculate, in general, analytically intractable complex integrals of posterior density functions. In the Monte Carlo integration methods, the posterior density function is approximated by the following empirical estimate

$$p(\mathbf{x}_k|\mathbf{Y}_k) \approx \hat{p}(\mathbf{x}_k|\mathbf{Y}_k) \triangleq \frac{1}{N}\sum_{i=1}^{N} \delta(\ \mathbf{x}_k - \mathbf{x}_k^i\ ), \tag{2.120}$$

where $\{\mathbf{x}_k^i, i = 1, \cdots, N\}$ are the N independent and identically distributed (i.i.d.) random samples (or particles), drawn from $p(\mathbf{x}_k|\mathbf{Y}_k)$, at time k, and $\delta(\cdot)$ denotes the Dirac delta function. When the number of particles N is sufficiently large, $\hat{p}(\mathbf{x}_k|\mathbf{Y}_k)$ approximates well the true posterior $p(\mathbf{x}_k|\mathbf{Y}_k)$. The posterior filtering density $p(\mathbf{x}_k|\mathbf{Y}_k)$ is the marginal of the joint posterior density $p(\mathbf{X}_k|\mathbf{Y}_k)$ whose empirical estimate can similarly be represented as follows:

$$p(\mathbf{X}_k|\mathbf{Y}_k) \approx \hat{p}(\mathbf{X}_k|\mathbf{Y}_k) \triangleq \frac{1}{N}\sum_{i=1}^{N} \delta(\ \mathbf{X}_k - \mathbf{X}_k^i\ ), \tag{2.121}$$

where $\{\mathbf{X}_k^i, i = 1, \cdots, N\}$ are the N i.i.d. random samples (or particles) drawn from the joint posterior distribution $p(\mathbf{X}_k|\mathbf{Y}_k)$.

With this approximation of the posterior, the averaged statistics or expectation of the vector-valued function $\mathbf{g}(\mathbf{x}_k)$ of the state can be approximated as follows:

$$E[\,\mathbf{g}(\mathbf{x}_k)\,] = \int \mathbf{g}(\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{Y}_k)\ d\mathbf{x}_k \tag{2.122}$$

$$\approx \int \mathbf{g}(\mathbf{x}_k)\ \hat{p}(\mathbf{x}_k|\mathbf{Y}_k)\ d\mathbf{x}_k \tag{2.123}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \int \mathbf{g}(\mathbf{x}_k)\ \delta(\ \mathbf{x}_k - \mathbf{x}_k^i\ )\ d\mathbf{x}_k \tag{2.124}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \mathbf{g}(\mathbf{x}_k^i) \triangleq \hat{E}_N[\mathbf{g}(\mathbf{x}_k)] \tag{2.125}$$

If $\mathbf{g}(\mathbf{x}_k) = \mathbf{x}_k$, we get the following optimal MMSE estimate

$$\hat{\mathbf{x}}_k = E[\,\mathbf{x}_k|\mathbf{Y}_k\,] = \int \mathbf{x}_k\; p(\mathbf{x}_k|\mathbf{Y}_k)\; d\mathbf{x}_k \approx \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_k^i. \tag{2.126}$$

By the Kolmogorov's *Strong Law of Large Numbers*, the sampling-based expectation $\hat{E}_N[\mathbf{g}(\mathbf{x}_k)]$ converges to the true expectation $E[\,\mathbf{g}(\mathbf{x}_k)\,]$ almost surely (a.s.) as N goes to infinity [81], i.e.,

$$\hat{E}_N[\mathbf{g}(\mathbf{x}_k)] \xrightarrow[N\to\infty]{a.s.} E[\,\mathbf{g}(\mathbf{x}_k)\,], \tag{2.127}$$

where $\xrightarrow{a.s.}$ denotes '*almost surely*' convergence. In addition, if the variance of $\mathbf{g}(\mathbf{x}_k)$ is bounded, i.e., $Var[\mathbf{g}(\mathbf{x}_k)] \triangleq \sigma^2_{\mathbf{g}(\mathbf{x}_k)} < \infty$, then the error convergence rate is observed by the following *Central Limit Theorem*

$$\sqrt{N}\left(\hat{E}_N[\,\mathbf{g}(\mathbf{x}_k)\,] - E[\,\mathbf{g}(\mathbf{x}_k)\,]\right) \underset{N\to\infty}{\Longrightarrow} \mathcal{N}\,(\,\mathbf{0},\; Var[\mathbf{g}(\mathbf{x}_k)]\,)\,, \tag{2.128}$$

where $\Longrightarrow$ denotes convergence in distribution. In general Monte Carlo simulations, samples are drawn from the joint posterior distribution $p(\mathbf{X}_k|\mathbf{Y}_k)$ instead of the filtering posterior distribution $p(\mathbf{x}_k|\mathbf{Y}_k)$. In this case, the expectation can be similarly approximated as follows:

$$E[\,\mathbf{g}(\mathbf{X}_k)\,] = \int \mathbf{g}(\mathbf{X}_k)\; p(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k \tag{2.129}$$

$$\approx \int \mathbf{g}(\mathbf{X}_k)\; \hat{p}(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k = \frac{1}{N}\sum_{i=1}^{N} \mathbf{g}(\mathbf{X}_k^i) \triangleq \hat{E}_N[\mathbf{g}(\mathbf{X}_k)]. \tag{2.130}$$

In the derivation of the particle filtering algorithm, the complete form of expectation using the joint posterior distribution $p(\mathbf{X}_k|\mathbf{Y}_k)$ is used.

2.5.2 Bayesian Importance Sampling

In the perfect Monte Carlo sampling method, a set of weighted samples (or particles), drawn directly from the true posterior distribution, is assumed to approximate the integrals of posterior density functions and consequently to approximate the integrals of expectations to discrete finite sums. However, in many nonlinear estimation

problems, it may not be possible to effectively generate samples directly from the true posterior distribution, which is usually multi-variate, complex-shaped, and not exactly known but only to a proportional constant. To circumvent the difficulty of direct sampling from the true posterior density functions, general Monte Carlo integration methods typically employ the importance sampling method in which samples are drawn from easier-to-implement importance (or proposal) density functions rather than the unknown or hard-to-sample true posterior density functions.

A. Importance Sampling

Since sampling directly from the true posterior density $p(\mathbf{X}_k|\mathbf{Y}_k)$ is rarely possible, we use the importance sampling concept in which samples are drawn from a known proposal distribution $q(\mathbf{X}_k|\mathbf{Y}_k)$ that has the same support with the true posterior density [100]. In this case, the averaged statistics or expectation of $\mathbf{g}(\mathbf{X}_k)$ can be

formulated using the proposal distribution.

$$
\begin{aligned}
E[\,\mathbf{g}(\mathbf{X}_k)\,] &= \int \mathbf{g}(\mathbf{X}_k)\; p(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k \\
&= \int \mathbf{g}(\mathbf{X}_k)\; \frac{p(\mathbf{X}_k|\mathbf{Y}_k)}{q(\mathbf{X}_k|\mathbf{Y}_k)}\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k \\
&= \int \mathbf{g}(\mathbf{X}_k)\; \frac{\frac{p(\mathbf{Y}_k|\mathbf{X}_k)p(\mathbf{X}_k)}{p(\mathbf{Y}_k)}}{q(\mathbf{X}_k|\mathbf{Y}_k)}\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k \quad \text{(by Bayes rule)} \\
&= \int \mathbf{g}(\mathbf{X}_k)\; \frac{\frac{p(\mathbf{Y}_k|\mathbf{X}_k)p(\mathbf{X}_k)}{q(\mathbf{X}_k|\mathbf{Y}_k)}}{p(\mathbf{Y}_k)}\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k \\
&= \int \mathbf{g}(\mathbf{X}_k)\; \frac{w_k(\mathbf{X}_k)}{p(\mathbf{Y}_k)}\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k \quad \text{(by definition)} \\
&= \frac{1}{p(\mathbf{Y}_k)} \int \mathbf{g}(\mathbf{X}_k)\; w_k(\mathbf{X}_k)\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k \\
&= \frac{\int \mathbf{g}(\mathbf{X}_k)\; w_k(\mathbf{X}_k)\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k}{\int p(\mathbf{Y}_k|\mathbf{X}_k)\; p(\mathbf{X}_k)\; d\mathbf{X}_k} \quad \text{(by marginalization and conditionalization)} \\
&= \frac{\int \mathbf{g}(\mathbf{X}_k)\; w_k(\mathbf{X}_k)\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k}{\int p(\mathbf{Y}_k|\mathbf{X}_k)\; p(\mathbf{X}_k)\; \frac{q(\mathbf{X}_k|\mathbf{Y}_k)}{q(\mathbf{X}_k|\mathbf{Y}_k)}\; d\mathbf{X}_k} \\
&= \frac{\int \mathbf{g}(\mathbf{X}_k)\; w_k(\mathbf{X}_k)\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k}{\int \frac{p(\mathbf{Y}_k|\mathbf{X}_k)\; p(\mathbf{X}_k)}{q(\mathbf{X}_k|\mathbf{Y}_k)} q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k} \\
&= \frac{\int \mathbf{g}(\mathbf{X}_k)\; w_k(\mathbf{X}_k)\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k}{\int w_k(\mathbf{X}_k)\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k} \quad \text{(by definition)} \\
&= \frac{E_{q(\mathbf{X}_k|\mathbf{Y}_k)}[\,\mathbf{g}(\mathbf{X}_k)\; w_k(\mathbf{X}_k)\,]}{E_{q(\mathbf{X}_k|\mathbf{Y}_k)}[\,w_k(\mathbf{X}_k)\,]}
\end{aligned}
\tag{2.131}
$$

As a result, we could express the expectation as a function of the proposal distribution rather than the true distribution:

$$
E[\,\mathbf{g}(\mathbf{X}_k)\,] = \frac{\int w_k(\mathbf{X}_k)\; \mathbf{g}(\mathbf{X}_k)\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k}{\int w_k(\mathbf{X}_k)\; q(\mathbf{X}_k|\mathbf{Y}_k)\; d\mathbf{X}_k} = \frac{E_{q(\mathbf{X}_k|\mathbf{Y}_k)}[\,w_k(\mathbf{X}_k)\; \mathbf{g}(\mathbf{X}_k)\,]}{E_{q(\mathbf{X}_k|\mathbf{Y}_k)}[\,w_k(\mathbf{X}_k)\,]}, \tag{2.132}
$$

where $E_{q(\mathbf{X}_k|\mathbf{Y}_k)}$ denotes the expectations using the proposal distribution $q(\mathbf{X}_k|\mathbf{Y}_k)$, and the *unnormalized importance weights* $w_k(\mathbf{X}_k)$ are defined as

$$
w_k(\mathbf{X}_k) \triangleq \frac{p(\mathbf{Y}_k|\mathbf{X}_k)\, p(\mathbf{X}_k)}{q(\mathbf{X}_k|\mathbf{Y}_k)} = \frac{p(\mathbf{X}_k, \mathbf{Y}_k)}{q(\mathbf{X}_k|\mathbf{Y}_k)}. \tag{2.133}
$$

B. Sequential Importance Sampling

By using the importance sampling method, we could convert the expectation (or averaged statistics) of $\mathbf{g}(\mathbf{X}_k)$ from elaborating the unknown or hard-to-use true posterior distribution $p(\mathbf{X}_k|\mathbf{Y}_k)$ to using the easy-to-implement proposal distribution $q(\mathbf{X}_k|\mathbf{Y}_k)$. During the derivation of expectation expressed by the proposal distribution, we introduced the unnormalized importance weights $w_k(\mathbf{X}_k)$. Since the unnormalized importance weights $w_k(\mathbf{X}_k)$ are expressed by the proposal distribution $q(\mathbf{X}_k|\mathbf{Y}_k)$, we need some statistics to manipulate the proposal distribution. Here, we introduce the following definition:

$$q(\mathbf{X}_k|\mathbf{Y}_k) \triangleq q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)\ q(\mathbf{X}_{k-1}|\mathbf{Y}_{k-1}) \tag{2.134}$$

When we introduce the assumptions that proposal distributions are also governed by the Bayesian statistics (similar to true posterior distributions), and the current state is not dependent on future observations (true for filtering problems), we can derive the above definition as follows:

$$\begin{aligned} q(\mathbf{X}_k|\mathbf{Y}_k) &= q(\mathbf{x}_k, \mathbf{X}_{k-1}|\mathbf{Y}_k) \\ &= q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)\ q(\mathbf{X}_{k-1}|\mathbf{Y}_k) \\ &= q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)\ q(\mathbf{X}_{k-1}|\mathbf{Y}_{k-1}) \end{aligned} \tag{2.135}$$

On the other hand, we can derive the following two relationships for the true probability distributions:

$$\begin{aligned} p(\mathbf{Y}_k|\mathbf{X}_k) &= p(\mathbf{y}_k, \mathbf{Y}_{k-1}|\mathbf{X}_k) \\ &= p(\mathbf{y}_k|\mathbf{Y}_{k-1}, \mathbf{X}_k)\ p(\mathbf{Y}_{k-1}|\mathbf{X}_k) \\ &= p(\mathbf{y}_k|\mathbf{x}_k)\ p(\mathbf{Y}_{k-1}|\mathbf{X}_{k-1}) \quad \text{(by Assumption 2.1.2)} \end{aligned} \tag{2.136}$$

$$p(\mathbf{X}_k) = p(\mathbf{x}_k, \mathbf{X}_{k-1}) = p(\mathbf{x}_k|\mathbf{X}_{k-1})\ p(\mathbf{X}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})\ p(\mathbf{X}_{k-1}) \tag{2.137}$$

The current observation is assumed to be dependent only on the current state (Assumption 2.1.2) in the first derivation, and the states are assumed to be a first-order Markov process (Assumption 2.1.1) in the second derivation. By substituting

Eqs. (2.134), (2.136), and (2.137) into Eq. (2.133), we have

$$\begin{aligned} w_k(\mathbf{X}_k) &= \frac{p(\mathbf{Y}_k|\mathbf{X}_k)\, p(\mathbf{X}_k)}{q(\mathbf{X}_k|\mathbf{Y}_k)} \\ &= \frac{p(\mathbf{y}_k|\mathbf{x}_k)\; p(\mathbf{Y}_{k-1}|\mathbf{X}_{k-1})\, p(\mathbf{x}_k|\mathbf{x}_{k-1})\; p(\mathbf{X}_{k-1})}{q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)\; q(\mathbf{X}_{k-1}|\mathbf{Y}_{k-1})} \\ &= \frac{p(\mathbf{Y}_{k-1}|\mathbf{X}_{k-1})\, p(\mathbf{X}_{k-1})}{q(\mathbf{X}_{k-1}|\mathbf{Y}_{k-1})}\; \frac{p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{x}_{k-1})}{q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)} \\ &= w_{k-1} \frac{p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{x}_{k-1})}{q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)}. \end{aligned} \tag{2.138}$$

Therefore, we have the following recursive update equation for the unnormalized importance weights $w_k(\mathbf{X}_k)$ expressed by the state transition pdf $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, the likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$, and the proposal distribution $q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)$.

$$w_k(\mathbf{X}_k) = w_{k-1} \frac{p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{x}_{k-1})}{q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)} \tag{2.139}$$

With this equation, once an appropriate choice of the proposal distribution ($q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)$) is given, we can sequentially update the importance weights.

The proper choice of the proposal distribution $q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)$ is the most important design factor. The first intuitive choice is as follows:

$$q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k) = p(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k). \tag{2.140}$$

This is the optimal proposal distribution that minimizes the variance on the importance weights [22]. In this case, the proposal distribution is chosen by the true conditional state density given the previous state history and all observations. Sampling from this true posterior distribution is impractical. The second choice of the proposal distribution is

$$q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k). \tag{2.141}$$

The most popular choice of the proposal distribution is the transition prior:

$$q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}). \tag{2.142}$$

If an additive Gaussian process noise model is used, the proposal transition is

$$q(\mathbf{x}_k|\mathbf{X}_{k-1},\mathbf{Y}_k) = p(\mathbf{x}_k|\mathbf{X}_{k-1},\mathbf{Y}_k) = \mathcal{N}(\ \mathbf{f}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_{k-1},0),\ \mathbf{Q}_{k-1}\ ). \tag{2.143}$$

C. Expectation Calculation Using Sequential Importance Sampling

The expectation of any vector-valued nonlinear function $\mathbf{g}(\mathbf{X}_k)$ could be represented by the expectations of the proposal distribution that includes the unnormalized importance weights. Moreover, the unnormalized importance weights could be recursively updated with the knowledge of state transition density, likelihood, and proposal density. From these two information, we are now equipped with the mechanism to approximate the expectation of $\mathbf{g}(\mathbf{X}_k)$ by using the samples (or particles) drawn from the proposal distribution and to recursively update the importance weights. Let $\{\mathbf{X}_k^i, i = 1, 2, \cdots, N\}$ be N i.i.d. samples/particles at time instant k drawn from the proposal distribution $q(\mathbf{x}_k|\mathbf{X}_{k-1},\mathbf{Y}_k)$, where $\mathbf{X}_k^i = \mathbf{x}_{0:k}^i$ is the i-th sample/particle trajectory (from time $t = t_0$ to $t = t_k$) drawn from the proposal distribution $q(\mathbf{x}_k|\mathbf{X}_{k-1},\mathbf{Y}_k)$.

$$\begin{aligned}
E[\,\mathbf{g}(\mathbf{X}_k)\,] &= \frac{E_{q(\mathbf{X}_k|\mathbf{Y}_k)}[\,w_k(\mathbf{X}_k)\ \mathbf{g}(\mathbf{X}_k)\,]}{E_{q(\mathbf{X}_k|\mathbf{Y}_k)}[\,w_k(\mathbf{X}_k)\,]} \\
&= \frac{\int w_k(\mathbf{X}_k)\ \mathbf{g}(\mathbf{X}_k)\ q(\mathbf{X}_k|\mathbf{Y}_k)\ d\mathbf{X}_k}{\int w_k(\mathbf{X}_k)\ q(\mathbf{X}_k|\mathbf{Y}_k)\ d\mathbf{X}_k} \\
&\approx \frac{\frac{1}{N}\sum_{i=1}^{N} w_k(\mathbf{x}_{0:k}^i)\ \mathbf{g}(\mathbf{x}_{0:k}^i)}{\frac{1}{N}\sum_{i=1}^{N} w_k(\mathbf{x}_{0:k}^i)} \\
&= \sum_{i=1}^{N}\left(\frac{w_k(\mathbf{x}_{0:k}^i)}{\sum_{j=1}^{N} w_k(\mathbf{x}_{0:k}^j)}\right)\ \mathbf{g}(\mathbf{x}_{0:k}^i) \\
&= \sum_{i=1}^{N} \tilde{w}_k(\mathbf{x}_{0:k}^i)\ \mathbf{g}(\mathbf{x}_{0:k}^i)
\end{aligned} \tag{2.144}$$

As a result, we can approximate the expectation with the following weighted summation:

$$E[\,\mathbf{g}(\mathbf{X}_k)\,] = \sum_{i=1}^{N} \tilde{w}_k^i \; \mathbf{g}(\mathbf{x}_{0:k}^i), \tag{2.145}$$

where the *normalized importance weights* $\tilde{w}_k^i$ are defined as

$$\tilde{w}_k^i \triangleq \tilde{w}_k(\mathbf{x}_{0:k}^i) = \frac{w_k(\mathbf{x}_{0:k}^i)}{\sum\limits_{j=1}^{N} w_k(\mathbf{x}_{0:k}^j)} = \frac{w_k^i}{\sum\limits_{j=1}^{N} w_k^j}. \tag{2.146}$$

In the case of filtering problems, we do not need to keep the whole history of the sample trajectories; only the current set of samples/particles at time instant k is needed to calculate expectations. In this case, we have the following expressions for the approximated expectation and the normalized importance weights:

$$E[\,\mathbf{g}(\mathbf{x}_k)\,] \approx \sum_{i=1}^{N} \tilde{w}_k(\mathbf{x}_k^i) \; \mathbf{g}(\mathbf{x}_k^i) = \sum_{i=1}^{N} \tilde{w}_k^i \; \mathbf{g}(\mathbf{x}_k^i), \tag{2.147}$$

where the normalized importance weights w_k^i are defined as

$$\tilde{w}_k^i = \tilde{w}_k(\mathbf{x}_k^i) = \frac{w_k(\mathbf{x}_k^i)}{\sum\limits_{j=1}^{N} w_k(\mathbf{x}_k^j)} = \frac{w_k^i}{\sum\limits_{j=1}^{N} w_k^j}, \tag{2.148}$$

and $\{\mathbf{x}_k^i, i = 1, \cdots, N\}$ are the N i.i.d. random samples/particles at time instant k drawn from $q(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Y}_k)$.

For example, state estimate $\hat{\mathbf{x}}_k$ and its covariance $\mathbf{P}_k$ in the minimum mean squared error (MMSE) sense can be derived as follows:

$$\begin{aligned} \hat{\mathbf{x}}_k &= E[\,\mathbf{x}_k|\mathbf{Y}_k\,] = E_{p(\mathbf{x}_k|\mathbf{Y}_k)}[\,\mathbf{x}_k\,] = \int \mathbf{x}_k \; p(\mathbf{x}_k|\mathbf{Y}_k) \; d\mathbf{x}_k \\ &\approx \sum_{i=1}^{N} \tilde{w}_k(\mathbf{x}_k^i) \; \mathbf{x}_k^i = \sum_{i=1}^{N} \tilde{w}_k^i \; \mathbf{x}_k^i \end{aligned} \tag{2.149}$$

$$\begin{aligned} \mathbf{P}_k &= E[\,(\mathbf{x}_k - \hat{\mathbf{x}}_k)\,(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\,|\mathbf{Y}_k\,] = E_{p(\mathbf{x}_k|\mathbf{Y}_k)}[\,(\mathbf{x}_k - \hat{\mathbf{x}}_k)\,(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\,] \\ &= \int (\mathbf{x}_k - \hat{\mathbf{x}}_k)\,(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \; p(\mathbf{x}_k|\mathbf{Y}_k) \; d\mathbf{x}_k \approx \sum_{i=1}^{N} \tilde{w}_k^i \; \left(\mathbf{x}_k^i - \hat{\mathbf{x}}_k\right)\left(\mathbf{x}_k^i - \hat{\mathbf{x}}_k\right)^T \end{aligned} \tag{2.150}$$

D. Resampling

By recursively formulating the importance sampling concept for the general Monte Carlo integration method, we can reach the sequential importance sampling (SIS) algorithm that constitutes the fundamental skeleton of most sequential Monte Carlo methods developed over the last several decades. These sequential Monte Carlo methods are collectively referred to as particle filters in which a number of independent random state samples, called particles, are drawn from the proposal density functions and thus represent the posterior distributions. The associated weights of the particles are recursively updated by incorporating the state transition model, the observation model, and current measurements in the recursive Bayesian estimation framework. These SIS-based sequential Monte Carlo methods suffer from the well-known sampling degeneracy phenomenon, which means that the normalized weights of all particles except one become negligible as time increases (or recursive steps progress). This is because the variance of the importance weights increases stochastically over time (degeneracy) [100]. By introducing an additional sampling step (resampling step), Gordon et al. [33] discovered a new type of particle filter that could overcome the sampling degeneracy problem by employing the sampling importance resampling (SIR) algorithm. Due to the development of this effective algorithm and the increase of recent computing power, particle filters have become a practically implementable approach applicable to many difficult nonlinear estimation problems [18, 19, 23, 33, 100, 115]. Specifically, the resampling/selection stage can be summarized as follows:

1. Eliminate samples with low importance weights.

2. Multiply samples with high importance weights.

3. Associate to each particle $\mathbf{x}_k^i$ a number of children N_i ($\sum_i N_i = N$).

Some of the resampling research includes SIR (sampling-importance resampling), residual resampling, and minimum-variance sampling [100]. The SIR algorithm is

applied to the design of a new filter in the next section.

2.6 Extended Marginalized Particle Filter

While particle filters have many attractive features including their applicability to general nonlinear, non-Gaussian problems without approximations to noise probability distributions, they also suffer from some defects. The most serious defect is the increase of their computational cost in higher dimensional state-space models because a large number of samples are required in order to appropriately approximate the posterior distributions. One technique to surmount this problem without reducing the efficiency of sampling techniques is to reduce the dimension of the state space model by marginalizing out some of the state variable components. The concept of marginalization, also referred to as Rao-Blackwellization, is based on the Rao-Blackwellization theorem by which once some sufficient statistics Ψ for a state vector $\mathbf{x}$ are provided, then any estimate of $\mathbf{f}(\mathbf{x})$ can always be improved by conditioning on the sufficient statistics Ψ [12, 102]. Rao-Blackwellization, by taking out states partially for analytical or exact consideration, allows the basic approach to reduce the variance in estimation and thus to improve the efficiency of the estimation. Sampling-based marginalization was first introduced by Gelfand and Smith [30] for calculating the marginal probability densities in Monte Carlo sampling methods. Several researchers attempted to improve the performance of the generic particle filters by introducing this marginalization property in the marginalized particle filter or Rao-Blackwellized particle filter [31, 32, 37, 89, 109].

Marginalization (or Rao-Blackwellization) has generally been applied to the state variable components that can be expressed by linear dynamics with Gaussian process noise and thus can be handled by the linear Kalman filter. In this work, the idea of marginalizing out some state components could be effectively extended even to the state components that are expressed by nonlinear dynamics. While part of the state

components are represented by nonlinear dynamics with Gaussian process noise, those state components can be effectively marginalized out by employing the unscented Kalman filter (UKF) to deal with those state components. This idea utilizes the reasoning that the UKF can more accurately and effectively solve nonlinear estimation problems with Gaussian noise characteristics compared to the extended Kalman filter (EKF). Since most real-world problems are composed of high dimensional state-space models and, at the same time, generally can only be represented by highly nonlinear dynamics, incorporating the UKF to solve part of the nonlinear dynamics allows us to deal with many real-world problems in the particle filtering framework.

As one of the important applications, the vision-based tracking problem, which will be described in detail in the subsequent chapters, could be effectively handled by the current approach in the particle filtering framework. Since vision sensor measurements can be better represented by non-Gaussian noise characteristics, the vision-based tracking problem can be more effectively solved in the particle filtering framework. However, since the problem can only be completely described by a relatively high-dimensional state-space model, direct employment of the particle filter for this problem is almost impossible because an enormous number of samples are required to properly approximate the posterior distributions. Nevertheless, since the vision information itself directly provides position information only (and not directly but indirectly velocity and acceleration information over the progression of time), only the position state components with vision information measurements are solved in the particle filtering framework, and the other state components represented by nonlinear equations with Gaussian noise are handled by the UKF.

2.6.1 Marginalization or Rao Blackwellization

Let's decompose the states into two parts, the state components handled by the Kalman filters (including the EKF or UKF) and the components solved in the particle

filtering framework as follows:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^{kf} \\ \mathbf{x}_k^{pf} \end{bmatrix}, \tag{2.151}$$

where $\mathbf{x}_k^{kf}$ denotes the state components handled by the Kalman filtering framework and $\mathbf{x}_k^{pf}$ denotes the state components handled by the particle filtering framework.

Then, the Bayesian statistics allow us to decompose the posterior distribution into the relationships:

$$p(\mathbf{x}_k \mid \mathbf{Y}_k) = p(\mathbf{x}_k^{kf}, \mathbf{x}_k^{pf} \mid \mathbf{Y}_k) = \underbrace{p(\mathbf{x}_k^{kf} \mid \mathbf{x}_k^{pf}, \mathbf{Y}_k)}_{UKF \ or \ EKF} \ \underbrace{p(\mathbf{x}_k^{pf} \mid \mathbf{Y}_k)}_{PF} \tag{2.152}$$

or in general

$$p(\mathbf{x}_k^{kf}, \mathbf{X}_k^{pf} \mid \mathbf{Y}_k) = \underbrace{p(\mathbf{x}_k^{kf} \mid \mathbf{X}_k^{pf}, \mathbf{Y}_k)}_{UKF \ or \ EKF} \ \underbrace{p(\mathbf{X}_k^{pf} \mid \mathbf{Y}_k)}_{PF}, \tag{2.153}$$

where

$$\begin{aligned} p(\mathbf{X}_k^{pf}|\mathbf{Y}_k) &= \frac{p(\mathbf{y}_k|\mathbf{X}_k^{pf}, \mathbf{Y}_{k-1}) \ p(\mathbf{x}_k^{pf}|\mathbf{X}_{k-1}^{pf}, \mathbf{Y}_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \ p(\mathbf{X}_{k-1}^{pf}|\mathbf{Y}_{k-1}), \\ &= \frac{p(\mathbf{y}_k|\mathbf{x}_k^{pf}) \ p(\mathbf{x}_k^{pf}|\mathbf{x}_{k-1}^{pf})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \ p(\mathbf{X}_{k-1}^{pf}|\mathbf{Y}_{k-1}), \\ p(\mathbf{y}_k|\mathbf{Y}_{k-1}) &= \int p(\mathbf{y}_k|\mathbf{x}_k^{pf}) \ p(\mathbf{x}_k^{pf}|\mathbf{Y}_{k-1}) \ d\mathbf{x}_k^{pf}. \end{aligned} \tag{2.154}$$

Here, $p(\mathbf{x}_k^{kf}, \mathbf{X}_k^{pf} \mid \mathbf{Y}_k)$ is approximated by the UKF, $p(\mathbf{X}_k^{pf}|\mathbf{Y}_k)$ is recursively given by the particle filter, $p(\mathbf{x}_k^{pf}|\mathbf{x}_{k-1}^{pf})$ is the transition probability density of the particle filtering state components, and $p(\mathbf{y}_k|\mathbf{x}_k^{pf})$ is the measurement likelihood of the particle filtering state components.

Many real-world problems can be represented by the following state space equations in the continuous-time domain:

$$\dot{\mathbf{x}}^{pf} = \mathbf{A}^{pf}\mathbf{x}^{kf} + \mathbf{w}^{pf}, \tag{2.155}$$

$$\dot{\mathbf{x}}^{kf} = \mathbf{f}(\mathbf{x}^{kf}) + \mathbf{w}^{kf}, \tag{2.156}$$

$$\mathbf{y}^{pf} = \mathbf{h}(\mathbf{x}^{pf}) + \mathbf{v}. \tag{2.157}$$

This state space model in the continuous-time domain can be discretized into the following state space model in a discrete-time domain:

$$\mathbf{x}_{k+1}^{pf} = \mathbf{x}_k^{pf} + \mathbf{A}_k^{pf}\ \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}\ \mathbf{w}_k^{pf}, \tag{2.158}$$

$$\mathbf{x}_{k+1}^{kf} = \mathbf{f}_d(\mathbf{x}_k^{kf}) + \mathbf{G}_k^{kf}\ \mathbf{w}_k^{kf}, \tag{2.159}$$

$$\mathbf{y}_k^{pf} = \mathbf{h}(\mathbf{x}_k^{pf}) + \mathbf{v}_k, \tag{2.160}$$

where $\mathbf{A}_k^{pf} = \Delta t\ \mathbf{A}^{pf}$, $\mathbf{G}_k^{pf} = \Delta t$, $\mathbf{f}_d(\mathbf{x}_k^{kf}) \cong \mathbf{x}_k^{kf} + \Delta t\ \mathbf{f}(\mathbf{x}_k^{kf})$, and $\mathbf{G}_k^{kf} = \Delta t$.
The following matrix form will also be used in order to apply the extended marginalized particle filter:

$$\mathbf{x}_{k+1}^{kf} = \mathbf{f}_d(\mathbf{x}_k^{kf}) + \mathbf{G}_k^{kf}\ \mathbf{w}_k^{kf}, \tag{2.161}$$

$$\mathbf{y}_k^{pf} = \mathbf{h}(\mathbf{x}_k^{pf}) + \mathbf{v}_k, \tag{2.162}$$

$$\mathbf{y}_k^{kf} = \mathbf{A}_k^{pf}\ \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}\ \mathbf{w}_k^{pf}, \text{ where } \mathbf{y}_k^{kf} = \mathbf{x}_{k+1}^{pf} - \mathbf{x}_k^{pf}. \tag{2.163}$$

Finally, the extended marginalized particle filter (EMPF) is applied to the following decomposed state space model:

Particle filter part:

$$\mathbf{x}_{k+1}^{pf} = \mathbf{x}_k^{pf} + \mathbf{A}_k^{pf}\ \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}\ \mathbf{w}_k^{pf}, \tag{2.164}$$

$$\mathbf{y}_k^{pf} = \mathbf{h}(\mathbf{x}_k^{pf}) + \mathbf{v}_k. \tag{2.165}$$

Kalman filter part:

$$\mathbf{x}_{k+1}^{kf} = \mathbf{f}_d(\mathbf{x}_k^{kf}) + \mathbf{G}_k^{kf}\ \mathbf{w}_k^{kf}, \tag{2.166}$$

$$\mathbf{y}_k^{kf} = \mathbf{A}_k^{pf}\ \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}\ \mathbf{w}_k^{pf}, \tag{2.167}$$

where $\mathbf{y}_k^{kf} = \mathbf{x}_{k+1}^{pf} - \mathbf{x}_k^{pf}$ is a pseudo-measurement from the particle filter part.

2.6.2 Extended Marginalized Particle Filter (EMPF)

The algorithm of the extended marginalized particle filter, that is the combination of the particle filter and the unscented Kalman filter, is summarized in Algorithm

2.1. The algorithm also includes the EKF part in parallel with the UKF part as an alternative to the UKF. If the nonlinear equations governing the state components of the Kalman filtering part are not so highly nonlinear and the Jacobian matrix of this part can be easily provided, this version of the EMPF might also be a choice which might have some advantage in the computational cost depending on specific applications.

Algorithm 2.1: Extended Marginalized Particle Filter (Combined Particle Filter and Unscented Kalman Filter Algorithm)

1. Initialization ($k = 0$)

For $i = 1, \cdots, N$, generate samples or particles $\mathbf{x}_0^{pf,i} \triangleq \mathbf{x}_{0|-1}^{pf,i} \sim p(\mathbf{x}_0^{pf})$, and set $\{\mathbf{x}_{0|-1}^{kf,i}, \mathbf{P}_{0|-1}^{i}\} = \{\bar{\mathbf{x}}_0^{kf}, \mathbf{P}_0^{kf}\}$.

2. Calculate the importance weights w_k^i and normalize:

$$w_k^i = p(\mathbf{y}_k|\mathbf{X}_k^{pf,i}, \mathbf{Y}_{k-1}) = p(\mathbf{y}_k|\mathbf{x}_k^{pf,i}),\ i = 1, \cdots, N, \tag{2.168}$$

$$\tilde{w}_k^i = \tilde{w}_k(\mathbf{x}_k^i) = \frac{w_k(\mathbf{x}_k^i)}{\sum_{j=1}^{N} w_k(\mathbf{x}_k^j)} = \frac{w_k^i}{\sum_{j=1}^{N} w_k^j},\ i = 1, \cdots, N. \tag{2.169}$$

3. Particle filter measurement update (Resampling)

Particle resampling is performed based on normalized importance weights $\tilde{w}_k^i$.

Resample N particles with replacement:

$$\mathbf{x}_{k|k}^{pf,i} \in \{\mathbf{x}_{k|k-1}^{pf,j}\}_{j=1}^{N},\ \text{where } p(\mathbf{x}_{k|k}^{pf,i} = \mathbf{x}_{k|k-1}^{pf,j}) = \tilde{w}_k^j. \tag{2.170}$$

4. Particle filter time update (Prediction, Time propagation)

Predict particles $\mathbf{x}_{k+1|k}^{pf,i},\ i = 1, \cdots, N$,

$$\mathbf{x}_{k+1}^{pf,i} \triangleq \mathbf{x}_{k+1|k}^{pf,i} \sim p(\mathbf{x}_{k+1|k}^{pf}|\mathbf{X}_k^{pf,i}, \mathbf{Y}_k), \tag{2.171}$$

where

$$p(\mathbf{x}_{k+1|k}^{pf}|\mathbf{X}_k^{pf,i}, \mathbf{Y}_k) = \mathcal{N}\left(\mathbf{x}_{k|k}^{pf,i} + \mathbf{A}_k^{pf}\ \hat{\mathbf{x}}_{k|k-1}^{kf,i},\ \mathbf{A}_k^{pf}\ \mathbf{P}_{k|k-1}\ (\mathbf{A}_k^{pf})^T + \mathbf{G}_k^{pf}\ \mathbf{Q}_k^{pf}\ (\mathbf{G}_k^{pf})^T\right). \tag{2.172}$$

5. Kalman filter pseudo-measurement update

1) Extended Kalman filter

$$\hat{\mathbf{x}}_{k|k}^{kf,i} = \hat{\mathbf{x}}_{k|k-1}^{kf,i} + \mathbf{K}_k \left[\mathbf{y}_k^{kf,i} - \mathbf{A}_k^{pf} \hat{\mathbf{x}}_{k|k-1}^{kf,i} \right], \quad \text{where } \mathbf{y}_k^{kf,i} = \mathbf{x}_{k+1}^{pf,i} - \mathbf{x}_k^{pf,i}, \tag{2.173}$$

$$\mathbf{P}_{k|k}^{i} = \left[\mathbf{I} - \mathbf{K}_k \mathbf{A}_k^{pf} \right] \mathbf{P}_{k|k-1}^{i} = \mathbf{P}_{k|k-1}^{i} - \mathbf{K}_k \mathbf{N}_k {\mathbf{K}_k}^T, \tag{2.174}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}^{i} \, (\mathbf{A}_k^{pf})^T \, \mathbf{N}_k^{-1} = \mathbf{P}_{k|k-1}^{i} \, (\mathbf{A}_k^{pf})^T \left[\mathbf{A}_k^{pf} \mathbf{P}_{k|k-1}^{i} (\mathbf{A}_k^{pf})^T + \mathbf{G}_k^{pf} \mathbf{Q}_k^{pf} (\mathbf{G}_k^{pf})^T \right]^{-1}. \tag{2.175}$$

2) Unscented Kalman filter

$$\mathbf{S}_{k|k-1} = \{ chol(\mathbf{P}_{k|k-1}^{i}) \}^T, \tag{2.176}$$

$$\mathbf{X}_{k|k-1} = [\ \hat{\mathbf{x}}_{k|k-1}^{kf,i} \quad \hat{\mathbf{x}}_{k|k-1}^{kf,i} + \gamma\, \mathbf{S}_{k|k-1} \quad \hat{\mathbf{x}}_{k|k-1}^{kf,i} - \gamma\, \mathbf{S}_{k|k-1}\ ], \tag{2.177}$$

$$\mathbf{Y}_{k|k-1} = \mathbf{A}_k^{pf} \, \mathbf{X}_{k|k-1}, \tag{2.178}$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2n} w_i^{(m)} \mathbf{Y}_{i,k|k-1}, \tag{2.179}$$

$$\mathbf{P}_y = \sum_{i=0}^{2n} w_i^{(c)} (\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T + \mathbf{G}_k^{pf} \mathbf{Q}_k^{pf} (\mathbf{G}_k^{pf})^T, \tag{2.180}$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} w_i^{(c)} (\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}^{kf,i})(\mathbf{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T, \tag{2.181}$$

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_y^{-1}, \tag{2.182}$$

$$\hat{\mathbf{x}}_{k|k}^{kf,i} = \hat{\mathbf{x}}_{k|k-1}^{kf,i} + \mathbf{K}_k \left(\mathbf{y}_k^{kf,i} - \hat{\mathbf{y}}_{k|k-1} \right), \quad \text{where } \mathbf{y}_k^{kf,i} = \mathbf{x}_{k+1}^{pf,i} - \mathbf{x}_k^{pf,i}, \tag{2.183}$$

$$\mathbf{P}_{k|k}^{i} = \mathbf{P}_{k|k-1}^{i} - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T. \tag{2.184}$$

6. Kalman filter time update

1) Extended Kalman filter

$$\hat{\mathbf{x}}_{k+1|k}^{kf,i} = \mathbf{f}_d(\hat{\mathbf{x}}_{k|k}^{kf,i}) = \hat{\mathbf{x}}_{k|k}^{kf,i} + \Delta t\ \mathbf{f}(\hat{\mathbf{x}}_{k|k}^{kf,i}), \tag{2.185}$$

$$\mathbf{P}_{k+1|k}^{i} = \mathbf{A}_k^{kf} \, \mathbf{P}_{k|k}^{i} \, (\mathbf{A}_k^{kf})^T + \mathbf{G}_k^{kf} \, \mathbf{Q}_k^{kf} \, (\mathbf{G}_k^{kf})^T, \tag{2.186}$$

where

$$\mathbf{A}_k^{kf} = \frac{\partial \mathbf{f}_d}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}^{kf,i}} = \mathbf{I} + \Delta t\ \mathbf{F}_k, \quad \mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}^{kf,i}}. \tag{2.187}$$

2) Unscented Kalman filter

$$\mathbf{S}_{k|k} = \{chol(\mathbf{P}_{k|k}^i)\}^T, \tag{2.188}$$

$$\mathbf{X}_{k|k} = [\ \hat{\mathbf{x}}_{k|k}^{kf} \quad \hat{\mathbf{x}}_{k|k}^{kf} + \gamma\ \mathbf{S}_{k|k} \quad \hat{\mathbf{x}}_{k|k}^{kf} - \gamma\ \mathbf{S}_{k|k}\ ], \tag{2.189}$$

$$\mathbf{X}_{k+1|k}^* = \mathbf{f}_d(\mathbf{X}_{k|k}), \tag{2.190}$$

$$\hat{\mathbf{x}}_{k+1|k}^{kf,i} = \sum_{i=0}^{2n} w_i^{(m)} \mathbf{X}_{i,k+1|k}^*, \tag{2.191}$$

$$\mathbf{P}_{k+1|k}^i = \sum_{i=0}^{2n} w_i^{(c)} (\mathbf{X}_{i,k+1|k}^* - \hat{\mathbf{x}}_{k+1|k}^{kf,i})(\mathbf{X}_{i,k+1|k}^* - \hat{\mathbf{x}}_{k+1|k}^{kf,i})^T + \mathbf{G}_k^{kf} \mathbf{Q}_k^{kf} (\mathbf{G}_k^{kf})^T. \tag{2.192}$$

CHAPTER 3

INTEGRATED NAVIGATION SYSTEM

3.1 Introduction

Recently, due to the easy availability of low-cost inertial measurement units (IMU's) consisting of accelerometer and rate gyro triads rigidly mounted on a vehicle, the inertial navigation system (INS) has become a backbone in cost-effective autonomous unmanned aerial vehicles (UAV's). The INS can provide the high-rate vehicle position, the velocity, and the attitude information necessary for flight guidance, navigation, and control (GNC) systems. The measured three-axis accelerations and angular rates from accelerometer and rate gyro triads, respectively, are integrated to obtain the vehicle's position, velocity, and attitude in an on-board computer. Since acceleration from the accelerometers and angular rate from the rate gyros are generally susceptible to various measurement noise sources, the accuracy of position/velocity and attitude information degrades with time [5, 7, 62, 103, 114]. Increasing the navigation accuracy by incorporating more accurate IMU measurement devices makes the INS system very expensive. On the other hand, recent research has shown that the growth of numerical errors in IMU navigation with time can be prevented by using low-cost absolute aiding sensors such as the Global Positioning System (GPS), a magnetometer, an altimeter, and so on [21, 64, 67, 119, 120, 129]. By combining the low-cost IMU with absolute aiding sensors, relatively accurate state estimate information can be provided to low-cost autonomous small UAV navigation systems. This integrated navigation system solves the time-degrading accuracy problem of INS by combining the short-term high-rate data characteristics of INS with the low rate but relatively time-independent accuracy of absolute aiding sensors. Since accelerometer

Figure 3.1: GTMax Research UAV

and rate gyro data (from an IMU) and aiding sensor measurements usually include various sources of noise, the accuracy of the navigation estimate is highly dependent on the details of noise filtering.

Typically, long-term INS errors are corrected by fusing IMU measurements with aiding sensor measurements such as GPS position/velocity, magnetometer heading information, and so on in the framework of the extended Kalman filter (EKF) [7, 64, 84, 120, 129]. Recently, the unscented Kalman filter (UKF) has drawn much research attention mainly because of its several salient features [53, 118, 124]. The UKF can be applied directly to nonlinear systems. Hence, it is not necessary to compute the Jacobian matrices which need to be provided in order to use an EKF. The EKF is based on the first-order gradient information of a nonlinear system, and thus its accuracy is known to be first order, while the UKF incorporates higher-order gradient information and is therefore known to be more accurate.

The purpose of this research is to develop an UKF-based integrated navigation system based on the existing EKF-based system well described in Ref. 20, 21, 45, 92. The developed algorithm details are targeted at being implemented on low-cost strapdown inertial navigation systems with an application to research UAV's [20, 21, 48, 49] shown in Figure 3.1.

This chapter starts off with Section 3.2 discussing the overall INS algorithm with

derivations of the detailed process models that will be used in the EKF/UKF-based navigation filter. The measurement models with specific sensor hardware characteristics are then presented in Section 3.3. The overall EKF and UKF algorithms with sequential measurement updates are summarized in Section 3.4. Simulation results with both EKF- and UKF-based navigation filters present the comparison of filter performance in Section 3.5. Finally, Section 3.6 summarizes the chapter.

3.2 Description of Inertial Navigation System

The combination of IMU hardware, composed of the accelerometer triad and the rate gyro triad, and navigation software, which includes the appropriate navigation algorithms, is called an inertial navigation system (INS). The INS can determine position, velocity, and attitude of a vehicle by measuring three-axis accelerations from an accelerometer triad and three-axis angular rates from a rate gyro triad. The fundamental part of the navigation algorithms is a navigation equation. The navigation equation for UAV local navigation can be expressed as follows [114]:

$$\frac{d^2\mathbf{r}^n}{dt^2} = \mathbf{a}^n_{cg} + \mathbf{g}^n. \tag{3.1}$$

This equation provides, with proper integration, the navigation quantities of velocity and position, thus it is called the navigation equation. Since we are developing the INS for autonomous research UAV flying in locally limited area, Earth curvature and Earth rotation effects are neglected (flat Earth and inertially fixed Earth assumption). These are reasonable assumptions for our purpose [129]. The velocity of a vehicle can be obtained by time-integrating the above navigation equation as

$$\mathbf{v}^n = \frac{d\mathbf{r}^n}{dt} \tag{3.2}$$

and its position, $\mathbf{r}^n$, can be obtained by integrating the equation twice. The accelerometer triad and the rate gyro triad in the strapdown INS, adopted in our UAV navigation system, are rigidly fixed to the vehicle. This means that the accelerometer

measurements or specific forces from the accelerometer triad are obtained in the body frame, hence they need to be transformed to the navigation frame in order to be used in the navigation equation.

$$\mathbf{a}_{cg}^{n} = \mathbf{C}_{b}^{n}\mathbf{a}_{cg}^{b}, \tag{3.3}$$

where $\mathbf{C}_{b}^{n}$ is the direction cosine matrix (DCM) from the body-fixed frame to the navigation frame. Then the navigation equation becomes

$$\frac{d^2\mathbf{r}^n}{dt^2} = \mathbf{C}_{b}^{n}\mathbf{a}_{cg}^{b} + \mathbf{g}^{n}. \tag{3.4}$$

The transformation matrix, $\mathbf{C}_{b}^{n}$, is a time-varying quantity and is a function of the current attitude. In order to determine the current attitude, we need an attitude evolution equation. Depending on which parameter set is used to express attitude, the attitude equation varies in forms. The two most popular ways to express attitude are using the three parameters of Euler angles ($\phi,\ \theta,\ \psi$) and the four parameters of a quaternion vector ($q_0,\ q_1,\ q_2,\ q_3$). Both ways have advantages and disadvantages. Using Euler angles is more intuitive for expressing the attitude but involves complex trigonometric relations in the attitude equation. On the other hand, the attitude equation is simple and without any trigonometric term when we use the four components of a quaternion as the attitude parameters. In our research, the quaternion set is used as the attitude parameters resulting in the following attitude equation:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}^b)\mathbf{q} = \frac{1}{2}\mathbf{Z}(\mathbf{q})\boldsymbol{\omega}^b, \tag{3.5}$$

where $\mathbf{\Omega}$ and $\mathbf{Z}$ are given below.

With the vehicle angular rate vector $\boldsymbol{\omega}^b(t)$ from the rate gyro triad and the initial quaternion vector $\mathbf{q}(t=0)$, this attitude equation can be integrated to get the quaternion vector $\mathbf{q}(t)$ or attitude at any time.

Combining the navigation equation, Eq. (3.4), expressed in state-space form by using

the vehicle velocity vector in Eq. (3.2) and the attitude equation, Eq. (3.5), we get the following ideal INS equations in the local level navigation frame mechanization:

$$\dot{\mathbf{r}}^n = \mathbf{v}^n, \tag{3.6}$$

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n(\mathbf{q})\ \mathbf{a}_{cg}^b + \mathbf{g}^n, \tag{3.7}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}^b)\ \mathbf{q} = \frac{1}{2}\mathbf{Z}(\mathbf{q})\ \boldsymbol{\omega}^b, \tag{3.8}$$

where

$$\mathbf{C}_b^n = (\mathbf{C}_n^b)^T = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix}, \tag{3.9}$$

$$\mathbf{\Omega}(\boldsymbol{\omega}^b) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^{bT} \\ \boldsymbol{\omega}^b & -\tilde{\boldsymbol{\omega}}^b \end{bmatrix} = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}, \quad \boldsymbol{\omega}^b = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \quad \tilde{\boldsymbol{\omega}}^b = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{3.10}$$

$$\mathbf{Z}(\mathbf{q}) = \begin{bmatrix} -\mathbf{q}_{13}^T \\ \tilde{\mathbf{q}}_{13} + q_0\ \mathbf{I}_{3\times 3} \end{bmatrix} = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} q_0 \\ \mathbf{q}_{13} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad \tilde{\mathbf{q}}_{13} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}, \tag{3.11}$$

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1, \quad \mathbf{g}^n = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T. \tag{3.12}$$

The magnitude of gravity at the surface of the WGS-84 ellipsoid can be expressed in the form [103]:

$$g = g_0 \frac{1 + g_1\ sin^2\phi_{lat}}{(1 - \epsilon^2\ sin^2\phi_{lat})^{\frac{1}{2}}}, \tag{3.13}$$

where ϕ_{lat} is the geodetic latitude, $g_0 = 9.7803267714\ m/sec^2$ is the gravity at

equator, $g_1 = 0.00193185138639\ m/sec^2$ is the gravity formula constant, and $\epsilon = 0.0818191908426$ is the first eccentricity.

The states of the ideal INS mechanization in Eq. (3.6) $\sim$ (3.8) include three components of positions, three components of velocities, and four parameters of a quaternion vector. The IMU is considered to provide three accelerations from the accelerometer triad and three angular rates from the rate gyro triad.

3.2.1 Continuous Process Model of INS Navigation

Since the extended Kalman filter is based on a continuous time process model in our research (Ref. 20, 21, 45, 48, 49), we first derive the continuous time process model for our integrated INS navigation system. The Ideal INS equations in Eq. (3.6) $\sim$ (3.8) are derived based on the assumption that the IMU is positioned at the vehicle's center of gravity and hence that the accelerometer triad measures acceleration at the center of gravity. Due to space limitation or installation convenience, the IMU is usually positioned at some position, $\mathbf{r}^b_{imu}$, relative to the vehicle's center of gravity. The acceleration vector at the IMU position, $\mathbf{a}^b$, can be calculated from the following relation:

$$\mathbf{a}^b = \mathbf{a}^b_{cg} + \Delta\mathbf{a}^b_{imu}, \tag{3.14}$$

where

$$\Delta\mathbf{a}^b_{imu} = \dot{\boldsymbol{\omega}}^b \times \mathbf{r}^b_{imu} + \boldsymbol{\omega}^b \times \left(\boldsymbol{\omega}^b \times \mathbf{r}^b_{imu}\right), \tag{3.15}$$

$\Delta\mathbf{a}^b_{imu}$ is the acceleration effect due to the IMU offset from the vehicle's center of gravity location, $\mathbf{r}^b_{imu}$. Since the rate gyro triad is rigidly fixed to the vehicle, angular rates at the IMU position are the same as those at vehicle's center of gravity. Hence, we do not differentiate between the two.

In general, the IMU sensor measurements are corrupted by various types of errors such as scale factors, misalignments, biases, and random noise [24, 28, 103, 129].

So the sensor models can incorporate these details depending on the required navigation accuracy and the inertial sensors at hand. In the INS development of our autonomous research UAV, it was enough to consider that true values are perturbed by two effects, a bias and a measurement noise in both accelerometer and rate gyro measurements [20, 119, 128].

$$\begin{aligned} \mathbf{a}_m^b &= \mathbf{a}^b + \mathbf{b}_a + \mathbf{n}_a, \\ \boldsymbol{\omega}_m^b &= \boldsymbol{\omega}^b + \mathbf{b}_\omega + \mathbf{n}_\omega \end{aligned} \tag{3.16}$$

or

$$\begin{aligned} \mathbf{a}^b &= \mathbf{a}_m^b - \mathbf{b}_a - \mathbf{n}_a = \bar{\mathbf{a}}^b - \mathbf{n}_a, \\ \boldsymbol{\omega}^b &= \boldsymbol{\omega}_m^b - \mathbf{b}_\omega - \mathbf{n}_\omega = \bar{\boldsymbol{\omega}}^b - \mathbf{n}_\omega, \end{aligned} \tag{3.17}$$

where $\bar{\mathbf{a}}^b = \mathbf{a}_m^b - \mathbf{b}_a$ and $\bar{\boldsymbol{\omega}}^b = \boldsymbol{\omega}_m^b - \mathbf{b}_\omega$ are bias corrected acceleration and angular rate vectors, respectively.

$\mathbf{a}^b$ and $\boldsymbol{\omega}^b$ are true acceleration and angular rate of the vehicle. $\mathbf{a}_m^b$ and ω_m^b are measured acceleration and measured angular rate from the IMU. $\mathbf{n}_a$ and $\mathbf{n}_\omega$ are the IMU acceleration and gyro rate measurement noise terms which are assumed to be zero-mean, white Gaussian noises.

By substituting Eq. (3.14)$\sim$(3.17) into Eq. (3.7), the velocity navigation equation becomes

$$\begin{aligned} \dot{\mathbf{v}}^n &= \mathbf{C}_b^n(\mathbf{q})\ \mathbf{a}_{cg}^b + \mathbf{g}^n = \mathbf{C}_b^n(\mathbf{q})\ \left(\mathbf{a}^b - \Delta\mathbf{a}_{imu}^b\right) + \mathbf{g}^n \\ &= \mathbf{C}_b^n(\mathbf{q})\ \left(\bar{\mathbf{a}}^b - \Delta\bar{\mathbf{a}}_{imu}^b\right) + \mathbf{g}^n - \mathbf{C}_b^n(\mathbf{q})\ \mathbf{n}_a, \end{aligned} \tag{3.18}$$

where

$$\Delta\bar{\mathbf{a}}_{imu}^b \cong \dot{\bar{\boldsymbol{\omega}}}_m^b \times \mathbf{r}_{imu}^b + \bar{\boldsymbol{\omega}}^b \times \left(\bar{\boldsymbol{\omega}}^b \times \mathbf{r}_{imu}^b\right). \tag{3.19}$$

For the approximation of the acceleration correction, $\Delta\bar{\mathbf{a}}_{imu}^b$, due to the IMU position offset from the center of gravity, the bias-corrected angular rate, $\bar{\boldsymbol{\omega}}^b$, and the low-pass

filtered[1] angular acceleration, $\dot{\boldsymbol{\omega}}^b_m$, are used. The additional noise terms that would appear in the expansion of $\Delta\mathbf{a}^b_{imu}$ are neglected in Eq. (3.18) on the assumption that they are small compared to the $\mathbf{n}_a$ term that appears due to the expansion of $\mathbf{a}^b$. This assumption is reasonable since $\Delta\mathbf{a}^b_{imu}$ is relatively small compared to $\mathbf{a}^b$ due to the small IMU position offset $\mathbf{r}^b_{imu}$ in our small UAV vehicle [115].

By substituting Eq. (3.17) into Eq. (3.8), the attitude equation becomes

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^b)\ \mathbf{q} = \frac{1}{2}\mathbf{Z}(\mathbf{q})\ \boldsymbol{\omega}^b = \frac{1}{2}\boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}^b)\ \mathbf{q} - \frac{1}{2}\mathbf{Z}(\mathbf{q})\ \mathbf{n}_\omega. \tag{3.20}$$

Here, we used the following relation:

$$\boldsymbol{\Omega}(\mathbf{x})\ \mathbf{q} = \mathbf{Z}(\mathbf{q})\ \mathbf{x}, \text{ for any vector } \mathbf{x} \in R^{3\times 1}. \tag{3.21}$$

In order to provide more fidelity of the accelerometer and rate gyro error models, time varying dynamics for acceleration biases and rate gyro biases are introduced. The biases are modeled as random walks with zero mean Gaussian driving terms in both the acceleration and rate gyro measurements.

$$\dot{\mathbf{b}}_a = \mathbf{n}_{b_a}, \tag{3.22}$$

$$\dot{\mathbf{b}}_\omega = \mathbf{n}_{b_\omega}, \tag{3.23}$$

where $\mathbf{n}_{b_a}$ and $\mathbf{n}_{b_\omega}$ are accelerometer and rate gyro bias noise which are considered to be zero-mean, white Gaussian.

Since we are interested in the sensor characteristics of low cost MEMS based IMUs such as the one used in the avionics of autonomous research UAV flight GNC system [21, 48], the accelerations and angular rates usually include large bias and scale factor errors. The scale factor effect can be considered as extra independent states [67, 80], or it can be included in the time-varying bias term that is sometimes

[1] In order to obtain angular acceleration, raw angular acceleration, obtained by using first-order difference (or Euler algorithm) of rate gyro measurements, is low-pass filtered with a time constant of $T_c = 0.1$ second.

used as a reasonable assumption [7,119] in our research. Because of the time varying bias effects, six additional states, three for acceleration bias errors and three for rate gyro bias errors, are needed for augmenting the state vector. Now, if we gather the position navigation equation (Eq. (3.6)), the velocity navigation equation (Eq. (3.18)), the attitude equation (Eq. (3.20)), the acceleration bias equation (Eq. (3.22)), and the rate gyro bias equation (Eq. (3.23)), we have the following INS navigation process model in continue-time state-space form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{G}(\mathbf{x}(t), t)\ \mathbf{w}(t) \tag{3.24}$$

or

$$\begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{\mathbf{q}} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ \mathbf{C}_b^n(\mathbf{q})\ (\ \bar{\mathbf{a}}^b - \Delta\bar{\mathbf{a}}_{imu}^b\ ) + \mathbf{g}^n \\ \frac{1}{2}\boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}^b)\mathbf{q} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{C}_b^n(\mathbf{q}) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & -\frac{1}{2}\mathbf{Z}(\mathbf{q}) & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{n}_r \\ \mathbf{n}_a \\ \mathbf{n}_\omega \\ \mathbf{n}_{b_a} \\ \mathbf{n}_{b_\omega} \end{bmatrix}, \tag{3.25}$$

where

$$\begin{aligned} \Delta\bar{\mathbf{a}}_{imu}^b &= \dot{\boldsymbol{\omega}}_m^b \times \mathbf{r}_{imu}^b + \bar{\boldsymbol{\omega}}^b \times \left(\bar{\boldsymbol{\omega}}^b \times \mathbf{r}_{imu}^b\right) \\ &= \begin{bmatrix} \dot{\omega}_2 z_a - \dot{\omega}_3 y_a + \bar{\omega}_1\left(\bar{\omega}_2 y_a + \bar{\omega}_3 z_a\right) - x_a\left(\bar{\omega}_2^2 + \bar{\omega}_3^2\right) \\ \dot{\omega}_3 x_a - \dot{\omega}_1 z_a + \bar{\omega}_2\left(\bar{\omega}_1 x_a + \bar{\omega}_3 z_a\right) - y_a\left(\bar{\omega}_1^2 + \bar{\omega}_3^2\right) \\ \dot{\omega}_1 y_a - \dot{\omega}_2 x_a + \bar{\omega}_3\left(\bar{\omega}_1 x_a + \bar{\omega}_2 y_a\right) - z_a\left(\bar{\omega}_1^2 + \bar{\omega}_2^2\right) \end{bmatrix}, \end{aligned} \tag{3.26}$$

$$\bar{\mathbf{a}}^b = \mathbf{a}_m^b - \mathbf{b}_a = [\ \bar{a}_1\ \bar{a}_2\ \bar{a}_3\ ]^T, \tag{3.27}$$

$$\bar{\boldsymbol{\omega}}^b = \boldsymbol{\omega}_m^b - \mathbf{b}_\omega = [\ \bar{\omega}_1\ \bar{\omega}_2\ \bar{\omega}_3\ ]^T. \tag{3.28}$$

The state vector of the INS process model is defined to include three position components, three velocity components, four quaternion components, three acceleration

biases, and three angular rate biases:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}^n \\ \mathbf{v}^n \\ \mathbf{q} \\ \mathbf{b}_a \\ \mathbf{b}_\omega \end{bmatrix}, \text{ where } \mathbf{r}^n = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \mathbf{v}^n = \begin{bmatrix} U \\ V \\ W \end{bmatrix}, \mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \mathbf{b}_a = \begin{bmatrix} a_{bx} \\ a_{by} \\ a_{bz} \end{bmatrix}, \mathbf{b}_\omega = \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix}. \tag{3.29}$$

Here, $\mathbf{r}^n$ and $\mathbf{v}^n$ are the position and the velocity vectors of the vehicle in the navigation frame, respectively. $\mathbf{q}$ is the quaternion vector that expresses the vehicle attitude. $\mathbf{b}_a$ and $\mathbf{b}_\omega$ are the IMU acceleration bias and the IMU rate gyro bias vectors, respectively. $\mathbf{n}_r$ is fictitious zero-mean, white noise associated with the position navigation equation.

In order to apply the EKF to the continuous-time process model, we need to calculate the following Jacobian matrix from the nonlinear system in Eq. (3.24) $\sim$ (3.28):

$$\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_k} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \left(\frac{\partial \mathbf{f}_v}{\partial \mathbf{q}}\right)_{3\times4} & \left(\frac{\partial \mathbf{f}_v}{\partial \mathbf{b}_a}\right)_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \left(\frac{\partial \mathbf{f}_q}{\partial \mathbf{q}}\right)_{4\times4} & \mathbf{0}_{4\times3} & \left(\frac{\partial \mathbf{f}_q}{\partial \mathbf{b}_\omega}\right)_{4\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}_k, \tag{3.30}$$

where

$$\left(\frac{\partial \mathbf{f}_v}{\partial \mathbf{q}}\right)_{3\times 4} =$$

$$\begin{bmatrix} 2(-q_3\bar{a}_2 + q_2\bar{a}_3) & 2(q_2\bar{a}_2 + q_3\bar{a}_3) & 2(-2q_2\bar{a}_1 + q_1\bar{a}_2 + q_0\bar{a}_3) & 2(-2q_3\bar{a}_1 - q_0\bar{a}_2 + q_1\bar{a}_3) \\ 2(q_3\bar{a}_1 - q_1\bar{a}_3) & 2(q_2\bar{a}_1 - 2q_1\bar{a}_2 - q_0\bar{a}_3) & 2(q_1\bar{a}_1 + q_3\bar{a}_3) & 2(q_0\bar{a}_1 - 2q_3\bar{a}_2 + q_2\bar{a}_3) \\ 2(-q_2\bar{a}_1 + q_1\bar{a}_2) & 2(q_3\bar{a}_1 + q_0\bar{a}_2 - 2q_1\bar{a}_3) & 2(-q_0\bar{a}_1 + q_3\bar{a}_2 - 2q_2\bar{a}_3) & 2(q_1\bar{a}_1 + q_2\bar{a}_2) \end{bmatrix},$$

$$\left(\frac{\partial \mathbf{f}_v}{\partial \mathbf{b}_a}\right)_{3\times 3} = -\mathbf{C}_b^n(\mathbf{q}), \quad \left(\frac{\partial \mathbf{f}_q}{\partial \mathbf{q}}\right)_{4\times 4} = \frac{1}{2}\mathbf{\Omega}(\bar{\boldsymbol{\omega}}^b), \quad \left(\frac{\partial \mathbf{f}_q}{\partial \mathbf{b}_\omega}\right)_{4\times 3} = -\frac{1}{2}\mathbf{Z}(\mathbf{q}). \tag{3.31}$$

Note that the IMU offset effect from the vehicle center of gravity, $\Delta\bar{\mathbf{a}}_{imu}^b$, and the change in gravity due to vehicle position, $\mathbf{g}^n$, are neglected in the calculation of the Jacobian matrix, $\mathbf{F}_k$. The first assumption is reasonable since the $\Delta\bar{\mathbf{a}}_{imu}^b$ term is relatively small compared to the $\bar{\mathbf{a}}^b$ term as discussed previously in the derivation of the velocity navigation equation in Eq. (3.18) [115]. In the local flight area of our small UAV, the gravity vector is usually assumed to be a constant vector $\mathbf{g}^n(\mathbf{r}^n) \simeq \text{const}$.

3.2.2 Discrete Process Model of INS Navigation

We need a discrete time process model in order to apply the unscented Kalman filter. The discrete time model of the continuous time process model in Eq. (3.24) $\sim$ (3.28)

can be approximated by the following first-order Euler integration algorithm:

$$\mathbf{r}_{k+1}^n = \mathbf{p}_k^n + \dot{\mathbf{r}}_k^n \, \Delta t = \mathbf{r}_k^n + \mathbf{v}_k^n \, \Delta t \tag{3.32}$$

$$\begin{aligned} \mathbf{v}_{k+1}^n &= \mathbf{v}_k^n + \dot{\mathbf{v}}_k^n \, \Delta t \\ &= \mathbf{v}_k^n + \left[\, \mathbf{C}_b^n(\mathbf{q}_k) \, (\, \bar{\mathbf{a}}_k^b - \Delta \bar{\mathbf{a}}_{imu,k}^b \,) + \mathbf{g}^n \, \right] \Delta t \; - \; \Delta t \, \mathbf{C}_b^n(\mathbf{q}_k) \, \mathbf{n}_{a,k} \end{aligned} \tag{3.33}$$

$$\begin{aligned} \mathbf{q}_{k+1} &= \mathbf{q}^{\frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_k^b) \, \Delta t} \, \mathbf{q}_k \simeq \left[\, \mathbf{I}_{4\times 4} + \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_k^b) \, \Delta t \, \right] \, \mathbf{q}_k \\ &= \left[\, \mathbf{I}_{4\times 4} + \frac{1}{2}\boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}_k^b - \mathbf{n}_{\omega,k}) \, \Delta t \, \right] \, \mathbf{q}_k \\ &= \left[\, \mathbf{I}_{4\times 4} + \frac{1}{2}\boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}_k^b) \, \Delta t \, \right] \, \mathbf{q}_k - \frac{\Delta t}{2} \, \boldsymbol{\Omega}(\mathbf{n}_{\omega,k}) \mathbf{q}_k \\ &= \left[\, \mathbf{I}_{4\times 4} + \frac{1}{2}\boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}_k^b) \, \Delta t \, \right] \, \mathbf{q}_k - \frac{\Delta t}{2} \, \mathbf{Z}(\mathbf{q}_k) \, \mathbf{n}_{\omega,k} \\ &\left(\because \boldsymbol{\Omega}(\mathbf{x}) \, \mathbf{q} = \mathbf{Z}(\mathbf{q}) \, \mathbf{x} \text{ for any } \mathbf{x} \in R^{3\times 1} \, \right) \end{aligned} \tag{3.34}$$

$$\mathbf{b}_{a,k+1} = \mathbf{b}_{a,k} + \mathbf{n}_{b_a,k} \, \Delta t \tag{3.35}$$

$$\mathbf{b}_{\omega,k+1} = \mathbf{b}_{\omega,k} + \mathbf{n}_{b_\omega,k} \, \Delta t \tag{3.36}$$

If we gather the above equations in a matrix form with the noise terms treated separately, we have the following approximate discrete nonlinear equation for the INS navigation process model.

$$\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, k) + \mathbf{G}_k \, \mathbf{w}_k \tag{3.37}$$

or

$$\begin{bmatrix} \mathbf{r}^n_{k+1} \\ \mathbf{v}^n_{k+1} \\ \mathbf{q}_{k+1} \\ \mathbf{b}_{a,k+1} \\ \mathbf{b}_{\omega,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^n_k + \mathbf{v}^n_k \, \Delta t \\ \mathbf{v}^n_k + \left[\, \mathbf{C}^n_b(\mathbf{q}_k) \, (\, \bar{\mathbf{a}}^b_k - \Delta \bar{\mathbf{a}}^b_{imu,k} \,) + \mathbf{g}^n \, \right] \Delta t \\ \left[\, \mathbf{I}_{4\times 4} + \frac{1}{2} \mathbf{\Omega}(\bar{\boldsymbol{\omega}}^b_k) \, \Delta t \, \right] \, \mathbf{q}_k \\ \mathbf{b}_{a,k} \\ \mathbf{b}_{\omega,k} \end{bmatrix}$$

$$+ \begin{bmatrix} \Delta t \, \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & -\Delta t \, \mathbf{C}^n_b(\mathbf{q}_k) & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} & -\frac{\Delta t}{2} \mathbf{Z}(\mathbf{q}_k) & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \Delta t \, \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \Delta t \, \mathbf{I}_{3\times 3} \end{bmatrix} \begin{bmatrix} \mathbf{n}_{r,k} \\ \mathbf{n}_{a,k} \\ \mathbf{n}_{\omega,k} \\ \mathbf{n}_{b_a,k} \\ \mathbf{n}_{b_\omega,k} \end{bmatrix} \tag{3.38}$$

where

$\bar{\mathbf{a}}^b_k = \mathbf{a}^b_{m,k} - \mathbf{b}_{a,k}$ and $\bar{\boldsymbol{\omega}}^b_k = \boldsymbol{\omega}^b_{m,k} - \mathbf{b}_{\omega,k}$ are bias-corrected IMU accelerometer and rate gyro measurements. Actual implementation in which the state estimate and the covariance matrix are propagated is given in Sections 2.4.4 and 3.4.2.

3.3 INS Navigation Measurement Model

In order to compensate for the data degradation of the low-cost IMU with time, an integrated navigation system that combines the low-cost IMU with aiding sensors is designed. Our autonomous research UAV is equipped with the following IMU and aiding sensors [20, 21, 48, 49].

- Inertial Sciences ISIS-IMU: three axis acceleration and angular rates
- NovAtel Millenium RT-2 Differential GPS receiver: inertial position and velocity
- Honeywell HMR-2300R Magnetometer: heading information

Table 3.1: Sensor Update Rates

Sensor	Update rate	Measurements
IMU	100 Hz	Vehicle acceleration & angular rates
DGPS	5 Hz	Inertial position & velocity
Magnetometer	20 Hz	Heading information
Vision sensor	10 Hz	Relative target position
Sonar altimeter	10 Hz	low altitude
Radar altimeter	10 Hz	high altitude

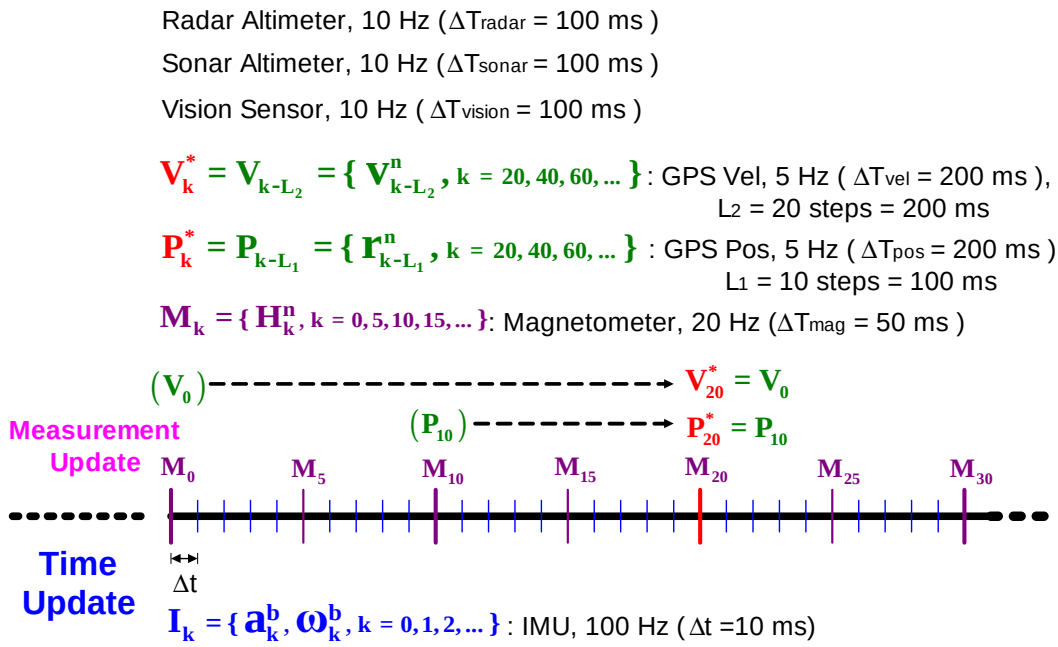


Figure 3.2: Sensor Update Timing

- Vision sensor: relative target position
- Custom-built Sonar with Polaroid 6500 ranging modules : short range altitude
- Roke Manor MRA Mk IV Radar Altimeter : long range altitude

Sensor update rates are summarized in Table 3.1, and various sensor update timings used in the simulation of the GTMax navigation system [20, 21] are shown in Figure 3.2[2]. Using the high-frequency vehicle accelerations and angular rates,

[2]Sonar and radar altimeters are excluded in this work since these two sensors are used to design an independent Kalman filter to estimate the terrain altitude in the GTMax navigation system.

the inertial navigation mechanization algorithm can provide the high-rate vehicle position/velocity and attitude necessary for an autonomous UAV flight GNC system. Because of IMU measurement errors, errors in the estimates of the vehicle position/velocity and attitude increase with time. In order to compensate for these long-term errors, aiding sensors such as the DGPS, the magnetometer, the altimeter, and so on are generally equipped in addition to the IMU. These aiding sensors measure data in a relatively low frequency but obtain data with time-independent accuracy. The IMU can provide complete navigation information such as the vehicle position/velocity and attitude if the IMU-measured accelerations and angular rates are time-integrated to get this navigation information. On the other hand, aiding sensors usually provide only partial navigation data, but the data can be directly used to get navigation information without involving dynamic equations. Since the IMU measurement rate is relatively higher than other aiding sensors, IMU measurements are considered to be a continuous data flow. Moreover, IMU-measured vehicle accelerations and angular rates are treated as the inputs to the process model.

Since aiding sensors have several different update rates, we need to carefully treat measurement updates in the Kalman filtering framework. Due to the variation in the number of available measurements depending on each time instant, the dimensions of the measurement vector and the Kalman gain matrix varies. In order to easily deal with this multi-rate sensor fusion problem, we apply the "sequential processing of measurement updates" method [2, 7, 13]. In this approach, measurement updates are not considered in their entirety or in big measurement vectors and matrices, but rather each measurement is treated separately, sequentially, and in several small-sized vectors and matrices. The addition of new aiding sensors is easier in this framework than in the standard Kalman filtering framework. Keeping this argument in mind, separate measurement models for all aiding sensors are described in this section.

3.3.1 DGPS Position and Velocity Measurement Model

Since the position and velocity of INS outputs degrade with time, a GPS receiver is augmented to frequently update the vehicle position and velocity in the navigation frame and to correct the long-term INS errors. The selected DGPS receiver provides position and velocity information at a slower rate than that of the INS system. Since the GPS antenna is mounted away from the vehicle's center of gravity, the DGPS sensor measures the position and velocity at the GPS mounting location with respect to the center of gravity, $\mathbf{r}^b_{gps}$. Furthermore, GPS measurements have latencies that need to be compensated. Considering this GPS latency, current position/velocity measurements are actually previous position/velocity, and current updates are based on older state estimates corresponding to this latency. In our integrated INS navigation, the GPS position and velocity have different latency, so their measurement updates are applied independently.

$$\mathbf{y}^1_k = \mathbf{h}^1(\mathbf{x}_k) + \mathbf{n}^{gps}_{r,k} \quad \Leftrightarrow \quad \mathbf{r}^{gps}_k = \mathbf{r}^n_{k-L1} + \mathbf{C}^n_b(\mathbf{q}_{k-L1})\, \mathbf{r}^b_{gps} + \mathbf{n}^{gps}_{r,k}, \tag{3.39}$$

$$\mathbf{y}^2_k = \mathbf{h}^2(\mathbf{x}_k) + \mathbf{n}^{gps}_{v,k} \quad \Leftrightarrow \quad \mathbf{v}^{gps}_k = \mathbf{v}^n_{k-L2} + \mathbf{C}^n_b(\mathbf{q}_{k-L2})\, \bar{\boldsymbol{\omega}}^b_{k-L2} \times \mathbf{r}^b_{gps} + \mathbf{n}^{gps}_{v,k}, \tag{3.40}$$

where $\mathbf{r}^n_{k-L1}$ and $\mathbf{v}^n_{k-L2}$ are the time-delayed vehicle position and vehicle velocity vectors in the navigation frame, and $\bar{\boldsymbol{\omega}}_{k-L2}$ is the time-delayed vehicle angular rate vector in the body frame. Time delay comes from the GPS sensor latency, $L_1 = \frac{GPS\ position\ latency}{\Delta t}$, and $L_2 = \frac{GPS\ velocity\ latency}{\Delta t}$. $\mathbf{r}^b_{gps}$ is the location of the GPS antenna relative to vehicle's center of gravity location in the body frame.

$$\mathbf{H}^1 = \frac{\partial \mathbf{h}^1}{\partial \mathbf{x}} = \begin{bmatrix} \left(\frac{\partial \mathbf{h}^1}{\partial \mathbf{r}^n}\right)_{3\times3} & \mathbf{0}_{3\times3} & \left(\frac{\partial \mathbf{h}^1}{\partial \mathbf{q}}\right)_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}, \tag{3.41}$$

$$\mathbf{H}^2 = \frac{\partial \mathbf{h}^2}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{v}^n}\right)_{3\times3} & \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{q}}\right)_{3\times4} & \mathbf{0}_{3\times3} & \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{b}_\omega}\right)_{3\times3} \end{bmatrix}, \tag{3.42}$$

where

$$\left(\frac{\partial \mathbf{h}^1}{\partial \mathbf{r}^n}\right)_{3\times3} = \mathbf{I}_{3\times3}, \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{v}^n}\right)_{3\times3} = \mathbf{I}_{3\times3}, \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{q}}\right)_{3\times4} = \mathbf{0}_{3\times4}, \quad \left(\frac{\partial \mathbf{h}^2}{\partial \mathbf{b}_\omega}\right)_{3\times3} = \mathbf{0}_{3\times3}, \tag{3.43}$$

$$\left(\frac{\partial \mathbf{h}^1}{\partial \mathbf{q}}\right)_{3\times 4} =$$

$$\begin{bmatrix} 2(-q_3y_g + q_2z_g) & 2(q_2y_g + q_3z_g) & 2(-2q_2x_g + q_1y_g + q_0z_g) & 2(-2q_3x_g - q_0y_g + q_1z_g) \\ 2(q_3x_g - q_1z_g) & 2(q_2x_g - 2q_1y_g - q_0z_g) & 2(q_1x_g + q_3z_g) & 2(q_0x_g - 2q_3y_g + q_2z_g) \\ 2(-q_2x_g + q_1y_g) & 2(q_3x_g + q_0y_g - 2q_1z_g) & 2(-2q_0x_g + q_3y_g - 2q_2z_g) & 2(q_1x_g + q_2y_g) \end{bmatrix}. \tag{3.44}$$

Note that the quaternion effect and the angular rate bias effect are approximated to be negligible in the Jacobian computation of the GPS velocity measurement.

3.3.2 Magnetometer Measurement Model

Three-axes magnetometer measures the Earth magnetic field in a body frame. The measured Earth magnetic field vector is used to correct for yaw angle or heading compensation. The basic idea for yaw angle compensation is as follows. The residual quantity in the measurement of a declination angle will be the same as the residual of a yaw angle. Hence, we first compute the residual value of the declination angle instead of that of the yaw angle, and then this residual (difference between the ideal declination angle given by the world magnetic model and the measured declination angle) is used to compensate for the yaw angle in the framework of the extended Kalman filter. The following measurement model is used for the yaw angle measurement:

$$\begin{aligned} &\mathbf{y}_k^3 = \mathbf{h}^3(\mathbf{x}_k) + \mathbf{n}_\psi \\ &\Leftrightarrow \quad \psi = atan2\left(C_{21},\ C_{11}\right) + \mathbf{n}_\psi = atan2\left[2(q_1q_2 + q_0q_3),\ q_0^2 + q_1^2 - q_2^2 - q_3^2\right] + \mathbf{n}_\psi, \end{aligned} \tag{3.45}$$

where ψ is the true heading angle, and C_{ij} are the corresponding components in the rotation matrix $\mathbf{C}_b^n$ expressed in (3.9). The measurement Jacobian matrix for the magnetometer measurement $\mathbf{H}^3$ is computed using these relations.

$$\mathbf{H}^3 = \frac{\partial \mathbf{h}^3}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \left(\frac{\partial \mathbf{h}^3}{\partial \mathbf{q}}\right)_{1\times 4} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}, \tag{3.46}$$

where

$$\left(\frac{\partial \mathbf{h}^3}{\partial \mathbf{q}}\right)_{1\times 4} = \begin{bmatrix} \frac{\partial \psi}{\partial q_0} & \frac{\partial \psi}{\partial q_1} & \frac{\partial \psi}{\partial q_2} & \frac{\partial \psi}{\partial q_3} \end{bmatrix}, \tag{3.47}$$

$$\frac{\partial \psi}{\partial q_0} = s_x(q_3 C_{11} - q_0 C_{21}), \quad \frac{\partial \psi}{\partial q_1} = s_x(q_2 C_{11} - q_1 C_{21}), \tag{3.48}$$

$$\frac{\partial \psi}{\partial q_2} = s_x(q_1 C_{11} + q_2 C_{21}), \quad \frac{\partial \psi}{\partial q_3} = s_x(q_0 C_{11} + q_3 C_{21}), \tag{3.49}$$

$$s_x = \frac{2}{(C_{11})^2 + (C_{21})^2}. \tag{3.50}$$

Note that both the heading angle expression and the measurement Jacobian matrix for the magnetometer triad $\mathbf{H}^3$ do not explicitly depend on the magnetometer triad. The magnetometer triad is involved in the calculation of the yaw angle residual that is used to compensate for the heading angle. Computation of the yaw angle residual will be explained in detail in the Section 3.4.1.

3.3.3 Quaternion Norm Pseudo-Measurement Model

In order to improve the quaternion norm characteristics, the following pseudo-measurement model is added:

$$\begin{aligned} &\mathbf{y}_k^4 = \mathbf{h}^4(\mathbf{x}_k) + \mathbf{n}_{qnorm} \\ &\Leftrightarrow \quad \|\mathbf{q}\|^2 = q_0^2 + q_1^2 - q_2^2 - q_3^2 + \mathbf{n}_{qnorm}. \end{aligned} \tag{3.51}$$

$$(\mathbf{y}_k^4)_{measurement} = 1. \tag{3.52}$$

The measurement Jacobian matrix for the pseudo-measurement, $\mathbf{H}^4$, is computed using the following relation:

$$\mathbf{H}^4 = \frac{\partial \mathbf{h}^4}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \left(\frac{\partial \mathbf{h}^4}{\partial \mathbf{q}}\right)_{1\times 4} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}, \tag{3.53}$$

where

$$\left(\frac{\partial \mathbf{h}^4}{\partial \mathbf{q}}\right)_{1\times 4} = \begin{bmatrix} 2\,q_0 & 2\,q_1 & 2\,q_2 & 2\,q_3 \end{bmatrix}. \tag{3.54}$$

3.4 Integrated INS Navigation Using EKF and UKF

In order to fuse aiding sensor measurements with the INS navigation algorithm, both the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) are used. This section describes the sensor fusion architecture and the details of Kalman filter implementation. Figure 3.3 shows the overall integrated strapdown INS mechanization. Note that only bias effects are considered to be independent states and are used to correct for IMU sensor drift.

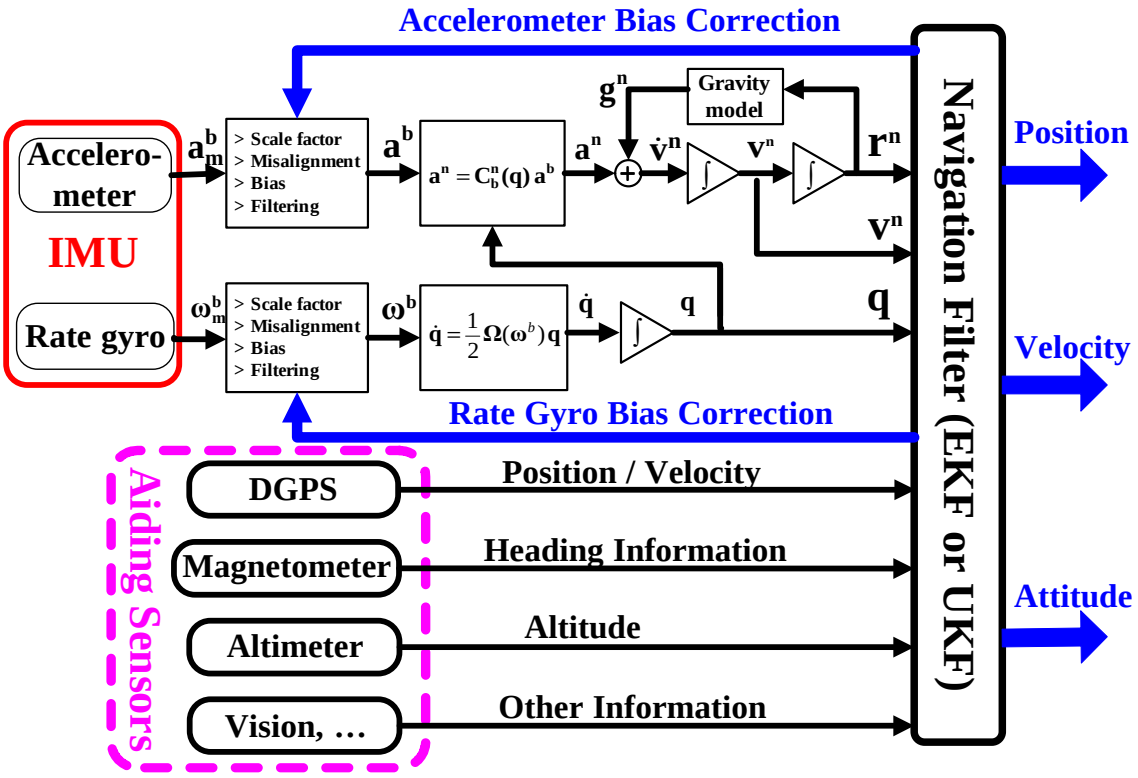


Figure 3.3: Integrated Strapdown Inertial Navigation System.

The INS outputs are used as a reference trajectory and attitude. Aiding sensor measurements such as the vehicle inertial position and velocity from the DGPS and heading information from the three-axis magnetometer are applied to update the states of this trajectory and attitude, and hence limit long-term error growth with time.

3.4.1 Extended Kalman Filter with Sequential Measurement Updates

In order to design the baseline integrated INS system discussed in detail in the Ref. [20, 21,45], the extended Kalman filter is applied to the nonlinear continuous-time process model and the discrete-time measurement model in state-space form. As explained in the previous section, measurement models are expressed in separate forms for all sensors in order to apply the "Sequential processing of multi-rate measurements" [2, 7,111] described in section 2.2.2.

A. GPS Measurement Update with Sensor Latency Compensation

In (2.46), we need to use a proper state vector for the calculation of the nonlinear measurement model in case of sensor latency. Since the GPS position and velocity measurement models need to use the latency-compensated state vector (i.e., previous states depending on the corresponding latency), measurement estimate components $\hat{\mathbf{y}}_k^l,\ (l = 1, 2)$ that correspond to the GPS position and velocity are computed by using latency compensated states.

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l \left[\mathbf{y}_k^l - \hat{\mathbf{y}}_k^l \right],\ l = 1, 2, \tag{3.55}$$

where

$$\hat{\mathbf{y}}_k^l = \mathbf{h}^l(\hat{\mathbf{x}}_{k-L}^-), \tag{3.56}$$

and $L = \frac{Aiding\ sensor\ latency}{\Delta t}$ represents time-delayed steps corresponding to sensor latency ($L = L_1$ for GPS position latency and $L = L_2$ for GPS velocity latency.)

B. Magnetometer Measurement Update

A three-axis magnetometer measures the Earth magnetic field in a body frame. The measured Earth magnetic field vector is used to compensate for vehicle heading. The yaw angle measurement update is based on the assumption that the residual in the measurement of the declination angle $\Delta\psi_d$ will be the same as the residual of the

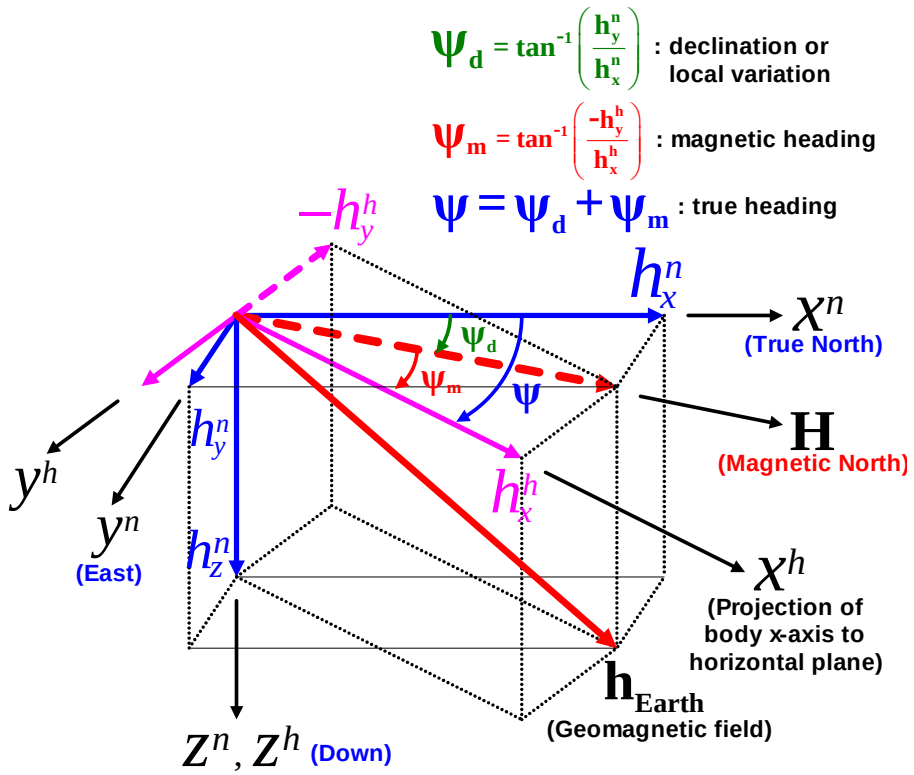


Figure 3.4: Earth Magnetic Field and Heading Angles

yaw angle $\Delta\psi$. Then we can use the residual value of the declination angle for the compensation of the yaw angle in the extended Kalman filter framework as follows:

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l \left[\mathbf{y}_k^l - \hat{\mathbf{y}}_k^l \right], \; l = 3, \tag{3.57}$$

where

$$\Delta\psi = \Delta\psi_d = \mathbf{y}_k^l - \hat{\mathbf{y}}_k^l \text{ (shifting to } -\pi \sim \pi), \tag{3.58}$$

$$\mathbf{y}_k^l = \psi_{d,WMM2005}, \tag{3.59}$$

$$\hat{\mathbf{y}}_k^l = \psi_{d,estimate} = atan2\left(h_y^n,\ h_x^n\right). \tag{3.60}$$

Here, ψ $(= \psi_d + \psi_m)$ is the true heading, ψ_d is the local variation or declination angle (shown in Figure 3.4), which is the angle between true north and magnetic north, ψ_m is the magnetic heading, and $\psi_{d,WMM2005}$ is obtained by the world magnetic model 2005 (WMM-2005) [87]. In addition, $\mathbf{h}^n = [\ h_x^n \ \ h_y^n \ \ h_z^n\ ]^T$ is the magnetic field triad

in the local navigation frame (NED) and is obtained by projecting the magnetic field vector on the local navigation frame (North-East-Down plane), using the last vehicle attitude estimate $\hat{\mathbf{q}}$, to eliminate the effects of magnetic dip. Since $\mathbf{h}^n = \mathbf{C}_b^n \, \mathbf{h}^b$ and $\mathbf{h}^b = [\, h_x^b \;\; h_y^b \;\; h_z^b \,]^T$ are measured from the magnetometer triad, we get the following relations:

$$h_x^n = h_x^b \, C_{11} + h_y^b \, C_{12} + h_z^b \, C_{13}, \tag{3.61}$$

$$h_y^n = h_x^b \, C_{21} + h_y^b \, C_{22} + h_z^b \, C_{23}, \tag{3.62}$$

where C_{ij} are corresponding components in the rotation matrix $\mathbf{C}_b^n$ expressed in (3.9). Note that even though $\psi_{d,WMM2005}$ is not an actual measurement but is given by a model, it is used as if it were a measurement. We can think of it as a pseudo-measurement. On the other hand, the magnetometer measurement data are not used in the measurement equation but rather in the estimate equation since $\psi_{d,estimate}$ cannot be directly obtained from a measurement but is estimated by using part of the current state (here, quaternion in C_{ij}) and this measurement information.

C. Implementation

Overcoming the usual computational power limitations of a low-cost UAV onboard system required several efforts. First, we introduced the following noise definitions:

$$\mathbf{n}_v \triangleq -\mathbf{C}_b^n(\mathbf{q}) \, \mathbf{n}_a, \tag{3.63}$$

$$\mathbf{n}_q \triangleq -\frac{1}{2}\mathbf{Z}(\mathbf{q}) \, \mathbf{n}_\omega. \tag{3.64}$$

Then, the INS navigation process model in continuous-time state-space form becomes the following rather simple form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{w}_1(t) \tag{3.65}$$

or

$$\begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{\mathbf{q}} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ \mathbf{C}_b^n(\mathbf{q}) \; (\; \bar{\mathbf{a}}^b - \Delta\bar{\mathbf{a}}_{imu}^b \;) + \mathbf{g}^n \\ \frac{1}{2}\mathbf{\Omega}(\bar{\boldsymbol{\omega}}^b)\mathbf{q} \\ \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_r \\ \mathbf{n}_v \\ \mathbf{n}_q \\ \mathbf{n}_{b_a} \\ \mathbf{n}_{b_\omega} \end{bmatrix}. \tag{3.66}$$

Expressing the INS process model in Eq. (3.25) is more intuitive as it is represented by direct sensor noise ($\mathbf{n}_a$, $\mathbf{n}_\omega$). The random noise characteristics of accelerometers and rate gyros ($\mathbf{n}_a$, $\mathbf{n}_\omega$) are relatively easily modeled from sensor characteristics and sensor test outputs. On the other hand, the INS process model expressed in Eq. (3.66) involves difficulty in modeling the noise characteristics of $\mathbf{n}_v$ and $\mathbf{n}_q$ even though it has a simpler form and thus is computationally more efficient. By using this process model expression, the error covariance propagation equation becomes

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\; \mathbf{P}(t) + \mathbf{P}(t)\; \mathbf{F}^T(\hat{\mathbf{x}}(t), t) + \mathbf{Q}(t), \tag{3.67}$$

where $E[\mathbf{w}_1(t)\mathbf{w}_1(\tau)^T] = \mathbf{Q}(t)\delta(t-\tau)$ and $\mathbf{Q}(t)$ is a 16×16 matrix. Comparing the original error covariance propagation equation,

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\; \mathbf{P}(t) + \mathbf{P}(t)\; \mathbf{F}^T(\hat{\mathbf{x}}(t), t) + \mathbf{G}(\mathbf{x}(t), t)\; \mathbf{Q}(t)\; \mathbf{G}^T(\mathbf{x}(t), t), \tag{3.68}$$

where $E[\mathbf{w}(t)\mathbf{w}(\tau)^T] = \mathbf{Q}(t)\delta(t-\tau)$ and $\mathbf{Q}(t)$ is a 15×15 matrix, the process model in Eq. (3.66) contributes in the computational efficiency of the navigation filter since it removes the multiplications of the 16×15 big matrix $\mathbf{G}(\mathbf{x})$. As $\mathbf{n}_v$ is just accelerometer noise in the navigation frame and $det(\mathbf{C}_b^n) = 1$ in Eq. (3.63), statistical characteristics of noise $\mathbf{n}_v$ can be modeled to have the same statistical characteristics as $\mathbf{n}_a$. The noise $\mathbf{n}_q$ in the $\dot{\mathbf{q}}$ equation is more difficult to characterize. Since the $\mathbf{Z}$ matrix is composed of quaternions that have values smaller than 1, the statistical characteristics of noise $\mathbf{n}_q$ can be approximately determined based on those of noise $\mathbf{n}_\omega$ and have been tuned

through flight tests. The process noise covariance matrix $\mathbf{Q}$ in the current filter is

$$\begin{aligned}
\mathbf{Q} &= diag[\ \mathbf{Q}_{n_r}, \mathbf{Q}_{n_v}, \mathbf{Q}_{n_q}, \mathbf{Q}_{n_{b_a}}, \mathbf{Q}_{n_{b_\omega}}\ ] \\
&= diag[\ 0.0\ (ft/s)^2,\ 0.0\ (ft/s)^2,\ 0.0\ (ft/s)^2, \\
&\qquad 0.01\ (ft/s^2)^2,\ 0.01\ (ft/s^2)^2,\ 0.01(ft/s^2)^2, \\
&\qquad 0.0001,\ 0.0001,\ 0.0001,\ 0.0001, \\
&\qquad 0.001\ (ft/s^2)^2,\ 0.001\ (ft/s^2)^2,\ 0.001\ (ft/s^2)^2, \\
&\qquad 0.00001\ (rad/s)^2,\ 0.00001\ (rad/s)^2,\ 0.00001\ (rad/s)^2\ ].
\end{aligned} \tag{3.69}$$

The measurement noise covariance matrix of GPS position $\mathbf{R}_{gpsPos} = diag[5^2, 5^2, 7^2]$ ft^2. Similarly, the measurement noise covariance matrix of GPS velocity $\mathbf{R}_{gpsVel} = diag[5^2, 5^2, 7^2]\ (ft/sec)^2$.

In the implementation of the time update in Eqs. (2.43) and (2.44), the state estimate is integrated with a modified Euler (or trapezoidal) integration algorithm and the error covariance matrix is updated using a first-order Euler integration algorithm:

$$\begin{aligned}
\hat{\mathbf{x}}_k^- &= \hat{\mathbf{x}}(t_{k-1}) + \int_{t_{k-1}}^{t_k^-} \mathbf{f}(\hat{\mathbf{x}}(t))\ dt \\
&\simeq \hat{\mathbf{x}}(t_{k-1}) + \frac{\Delta t}{2}\left[\ \mathbf{f}(\hat{\mathbf{x}}(t_{k-1})) + \mathbf{f}(\mathbf{x}_1(t_k^-))\ \right] \\
&= \mathbf{x}_1(t_k^-) + \frac{\Delta t}{2}\left[\ \mathbf{f}(\mathbf{x}_1(t_k^-)) - \mathbf{f}(\hat{\mathbf{x}}(t_{k-1}))\ \right],
\end{aligned} \tag{3.70}$$

$$\mathbf{P}_k^- = \mathbf{P}_{k-1} + \Delta t\ \left[\mathbf{F}(\hat{\mathbf{x}}(t_{k-1}), t_{k-1})\ \mathbf{P}(t_{k-1}) + \mathbf{P}(t_{k-1})\ \mathbf{F}^T(\hat{\mathbf{x}}(t_{k-1}), t_{k-1}) + \mathbf{Q}(t_{k-1})\right], \tag{3.71}$$

where $\mathbf{x}_1(t_k^-) = \hat{\mathbf{x}}(t_{k-1}) + \Delta t\ \mathbf{f}(\hat{\mathbf{x}}(t_{k-1}))$.

One more thing worth mentioning is about the advantage of sequential measurement update. Aiding sensors in INS mechanization are usually not correlated each other and hence sequential measurement update is possible. In this case, measurement update is computationally efficient since series of smaller matrix inversion is involved instead of one big matrix inversion in Eq. (2.45). Since aiding sensors have

several different update rates, we need to carefully treat measurement updates in Kalman filtering framework. Due to the variation in the number of available measurements depending on the time instant, the dimensions of the measurement vector and the Kalman gain matrix also vary. In the sequential measurement update approach, it is easier to deal with this multi-rate sensor fusion problem. Furthermore, the addition of new aiding sensors is easier in this framework than in the standard Kalman filtering framework. The error covariance matrix in Eq. (2.47) is updated by using the following De Vries-Joseph implementation in the measurement update step. This implementation reduces the computational complexity of the Joseph formulation by judicious rearrangement of the matrix expressions and reuse of intermediate results(See Ref. 34, p.259, Table 6.19):

$$\begin{aligned} T_1 &= \mathbf{P}^{-}\mathbf{H}^T, \\ T_2 &= \mathbf{H}\, T_1 + \mathbf{R}, \\ \mathbf{K} &= T_1\, T_2^{-1}, \\ T_3 &= \frac{1}{2}\mathbf{K}\, T_2 - T_1, \\ T_4 &= T_3\, \mathbf{K}^T, \\ \mathbf{P} &= \mathbf{P}^{-} + T_4 + T_4^T. \end{aligned} \tag{3.72}$$

3.4.2 Unscented Kalman Filter with Sequential Measurement Updates

The original unscented Kalman filter was first developed by Julier et al. [53, 58, 59] and diversified into several algorithms by Merwe et al. [116, 118, 119]. The details of the unscented Kalman filter with sequential measurement updates is presented in Section 2.4.4. The idea behind this is to recursively update the state estimate and the error covariance matrix by using a more accurate nonlinear filtering approach, the unscented Kalman filter, while maintaining the advantages of the sequential measurement updates mentioned in Section 2.3.2. By using the new UKF with sequential

measurement updates instead of the standard UKF algorithm, we can add the advantages of the sequential measurement update strategy. We can easily handle the multi-rate sensor fusion problem and sensor latency compensation. The addition of new aiding sensors is easier in this framework than in the standard UKF framework. Furthermore, we can remove the computation of messy Jacobian matrices while maintaining at least second-order estimation accuracy.

A. Sequential Measurement Update with Sensor Latency Compensation

In Eq. (2.178), we need to use proper sigma points in order to calculate the nonlinear measurement model. Since the GPS position and velocity measurement models use the latency compensated state vector (i.e., previous states depending on the corresponding latency), measurement estimate components $\mathbf{Y}_k^l$ corresponding to the GPS position and velocity are computed by using sigma points based on latency compensated states.

$$\mathbf{Y}_k^l = \mathbf{h}^l(\mathbf{X}_{k-L}^{l-1}),\ l = 1, 2, \tag{3.73}$$

where

$$\mathbf{X}_{k-L}^{l-1} = [\ \hat{\mathbf{x}}_{k-L}^- \quad \hat{\mathbf{x}}_{k-L}^- + \gamma\, \mathbf{S}_k^{l-1} \quad \hat{\mathbf{x}}_{k-L}^- - \gamma\, \mathbf{S}_k^{l-1}\ ], \tag{3.74}$$

and $L = \frac{Aiding\ sensor\ latency}{\Delta t}$ represents time-delayed steps corresponding to sensor latency.

B. Implementation

By introducing new noise definitions in Eq. (3.63) and (3.64), we have the following INS navigation process model in discrete-time state-space form:

$$\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \tag{3.75}$$

or

$$\begin{bmatrix} \mathbf{r}_{k+1}^n \\ \mathbf{v}_{k+1}^n \\ \mathbf{q}_{k+1} \\ \mathbf{b}_{a,k+1} \\ \mathbf{b}_{\omega,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_k^n + \mathbf{v}_k^n \, \Delta t \\ \mathbf{v}_k^n + \left[\, \mathbf{C}_b^n(\mathbf{q}_k) \, (\, \bar{\mathbf{a}}_k^b - \Delta \bar{\mathbf{a}}_{imu,k}^b \,) + \mathbf{g}^n \, \right] \Delta t \\ \left[\, \mathbf{I}_{4\times 4} + \frac{1}{2} \mathbf{\Omega}(\bar{\boldsymbol{\omega}}_k^b) \, \Delta t \, \right] \, \mathbf{q}_k \\ \mathbf{b}_{a,k} \\ \mathbf{b}_{\omega,k} \end{bmatrix} + \Delta t \begin{bmatrix} \mathbf{n}_{r,k} \\ \mathbf{n}_{v,k} \\ \mathbf{n}_{q,k} \\ \mathbf{n}_{b_a,k} \\ \mathbf{n}_{b_\omega,k} \end{bmatrix}, \quad (3.76)$$

where $\bar{\mathbf{a}}_k^b = \mathbf{a}_{m,k}^b - \mathbf{b}_{a,k}$ and $\bar{\boldsymbol{\omega}}_k^b = \boldsymbol{\omega}_{m,k}^b - \mathbf{b}_{\omega,k}$ are bias-corrected IMU accelerometer and rate gyro measurements.

In order to preserve the same second-order accuracy applied to the time propagation of the process model in the EKF-based navigation system shown in Eq. (3.70), the continuous-time process model (Eq. (3.66)) is directly integrated by using the same modified Euler (or trapezoidal) integration algorithm instead of using the above first-order Euler (or rectangular) discretization. That is, the second part of Eq. (2.190) is applied to the time propagation of the process model in the UKF-based navigation system.

3.5 Simulation Model and Filter Performance Simulation

Since we aim to develop a low-cost integrated strapdown inertial navigation system with an application to research UAV's [20, 21, 48, 49], a high-fidelity nonlinear 6-DOF simulation environment is used. The entire 6-DOF simulation code, developed with C/C++, is equipped with a neural-network-based adaptive nonlinear controller [47]. Both the EKF- and UKF-based integrated navigation systems are also developed in C/C++ and are analyzed in the nonlinear 6-DOF simulation environment. Various meaningful nonlinear trajectories can be generated in the 6-DOF simulation. This trajectory data, corrupted by various sources of noise based on sensor modeling, provide fruitful test conditions for the navigation system. This section, at first, briefly

describes some simulation models and then presents simulation results that show filter performance.

3.5.1 Simulation Model

In order to increase the reality of the simulation model, the following features are included in the simulation model.

A. IMU Simulation Model

Noticing the effect of helicopter blade rotation vibration on IMU sensor measurements during flight tests, the following shake vibration effects around 33 Hz are added to each channel of the IMU accelerometer and rate gyro measurements.

$$\begin{aligned} \Delta a^b_{shake} &= a_{shake}\ sin(\Omega_{shake}\ t), \\ \Delta \omega^b_{shake} &= \omega_{shake}\ sin(\Omega_{shake}\ t), \end{aligned} \tag{3.77}$$

where $a_{shake} = 0.1 ft/sec^2$ is the accelerometer shake magnitude, $\omega_{shake} = 0.18\ rad/sec$ is the rate gyro shake magnitude, and $\Omega_{shake} = 2\pi * 33\ rad/sec$ is the shake frequency. By adding the vibration effect to the IMU filter sensor models in (3.16), we have the following IMU simulation sensor models.

$$\begin{aligned} \mathbf{a}^b_m &= sat\{\ \mathbf{a}^b_{cg} + \Delta\mathbf{a}^b_{imu} + \Delta\mathbf{a}^b_{shake} + \mathbf{b}_a + \mathbf{n}_a, -a_{limit}, +a_{limit}\ \}, \\ \boldsymbol{\omega}^b_m &= sat\{\ \boldsymbol{\omega}^b + \Delta\boldsymbol{\omega}^b_{shake} + \mathbf{b}_\omega + \mathbf{n}_\omega, -\omega_{limit}, +\omega_{limit}\ \}, \end{aligned} \tag{3.78}$$

where $\Delta\mathbf{a}^b_{imu} = \dot{\boldsymbol{\omega}}^b \times \mathbf{r}^b_{imu} + \boldsymbol{\omega}^b \times \left(\boldsymbol{\omega}^b \times \mathbf{r}^b_{imu}\right)$ is the acceleration effect due to the IMU offset from the vehicle's center of gravity location, $\mathbf{r}^b_{imu}$, and $sat\{\mathbf{x}, -x_{limit}, +x_{limit}\}$ is a saturation function in which each component of $\mathbf{x}$ is saturated at $-x_{limit}$ or $+x_{limit}$ when it exceeds these limits otherwise it has no effect.

B. GPS Simulation Model

The first-order lag model is used for GPS position error, which is added into the true GPS position instead of a simple white Gaussian noise addition. The GPS position error model in the first-order lag model is

$$\dot{\mathbf{r}}_{error}(t) = -\frac{1}{T}\mathbf{r}_{error}(t) + \mathbf{w}(t), \tag{3.79}$$

where $\mathbf{w}(t)$ is white Gaussian noise with standard deviation (0.1, 0.1, 0.15) ft in each direction. This can be approximated into the following discrete form:

$$\mathbf{r}_{error,k+1} = \left(1 - \frac{\Delta t}{T}\right) \mathbf{r}_{error,k} + \Delta t \ \mathbf{w}_k, \tag{3.80}$$

where $\Delta t = 0.2$ sec is the update time step of GPS position error and $T = 10$ sec is the time constant of GPS position error. This GPS position error is added to the true position at the GPS antenna location and delayed for the simulation of GPS position sensor latency.

$$\mathbf{r}^{gps}_{k+L_1} = \mathbf{r}^n_k + \mathbf{C}^n_b(\mathbf{q}_k)\ \mathbf{r}^b_{gps} + \mathbf{r}_{error,k}, \tag{3.81}$$

where L_1 is the number of steps in GPS position latency explained previously. GPS velocity error is modeled as white Gaussian noise. This velocity error is added to the true velocity at the GPS antenna location and delayed by the amount of GPS velocity sensor latency as follows:

$$\mathbf{v}^{gps}_{k+L_2} = \mathbf{v}^n_k + \mathbf{C}^n_b(\mathbf{q}_k)\ \boldsymbol{\omega}^b_k \times \mathbf{r}^b_{gps} + \mathbf{v}_{error,k}, \tag{3.82}$$

where L_2 is the number of steps in GPS velocity latency and $\mathbf{v}_{error,k}$ is simple white Gaussian noise with standard deviation (0.03, 0.03, 0.05) ft/sec in each direction.

C. Magnetometer Simulation Model

For the reference magnetic model, the following first-order approximation is used in the World Magnetic Model 2005 (WMM-2005).

$$\begin{aligned}
h^n_x &= -\left(\frac{a}{r}\right)^3 \left\{\ g^0_1(t)\ cos\phi_{lat} + \left[\ g^1_1(t)\ cos\lambda_{long} + h^1_1(t)\ sin\lambda_{long}\ \right] sin\phi_{lat}\ \right\}, \\
h^n_y &= -\left(\frac{a}{r}\right)^3 \left[\ g^1_1(t)\ sin\lambda_{long} - h^1_1(t)\ cos\lambda_{long}\ \right], \\
h^n_z &= -2\left(\frac{a}{r}\right)^3 \left\{\ g^0_1(t)\ sin\phi_{lat} - \left[\ g^1_1(t)\ cos\lambda_{long} + h^1_1(t)\ sin\lambda_{long}\ \right] cos\phi_{lat}\ \right\},
\end{aligned} \tag{3.83}$$

where $g^0_1(t) = -29556.8 + 8.0(t - t_0)$, $g^1_1(t) = -1671.7 + 10.6(t - t_0)$, $h^1_1(t) = 5079.8 - 20.9(t - t_0)$, t_0 is the reference date of the model ($t_0 = 2005.0$), t is the decimal year (for example, Oct. 1, 2007 as $t \simeq 2007.75$), a (6371.2 km) is the standard Earth

magnetic reference radius, $(\phi_{lat}, \lambda_{long}, r)$ are the latitude, longitude, and radius in a spherical, geocentric reference frame. $\mathbf{h}^n = [\ h_x^n \ \ h_y^n \ \ h_z^n\ ]^T$ is the magnetic field triad in the local navigation frame (NED). This magnetic field triad is normalized by its magnitude and we will keep the same notation for the normalized quantity with a slight abuse of notation.

Using the above relations, the magnetic field in the body frame $\mathbf{h}^b = [\ h_x^b \ \ h_y^b \ \ h_z^b\ ]^T$ is modeled as follows:

$$\mathbf{h}^b = \mathbf{C}_n^b\ \mathbf{h}^n + \mathbf{n}_h, \tag{3.84}$$

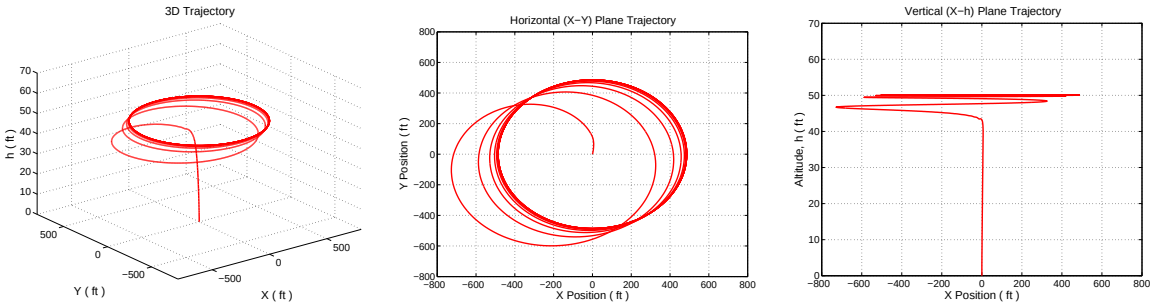
where $\mathbf{C}_n^b$ is the transformation matrix from the navigation frame to the body frame given in (3.9), $\mathbf{n}_h$ is the magnetic field noise in the body frame and is set to zero-mean, white-Gaussian noise with standard deviation 0.02 in normalized quantity.

D. Initial Conditions

Nominal values of the accelerometer bias and the rate gyro bias are all initialized to zero. In order to see the convergence of the navigation filters, true values of the biases are set to some nonzero values, $\mathbf{b}_{atrue}$ = (1.0, 1.0, 0.1) ft/sec^2 and $\mathbf{b}_{\omega true}$ = (0.1, 0.1, 0.1) rad/sec[3]

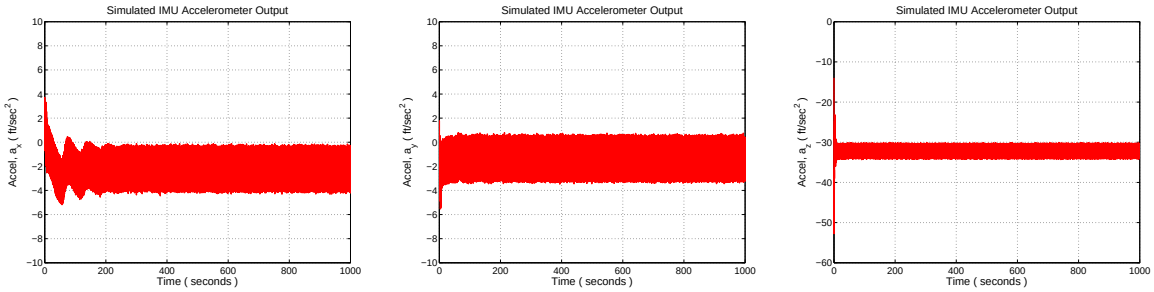
The initial position and velocity for both the true and the estimate states are (0, 0, 0) ft and (0, 0, 0) ft/sec, respectively. The initial attitude is (ϕ, θ, ψ) = (0,

[3]Considering the fact that the navigation system of the GTMax helicopter uses the accelerometer bias vector as $\mathbf{b}_a$ = (5.5, -3.0, 0) ft/sec^2 and the rate gyro bias as zero, these assumed values are not severe. The values at the GTMax navigation system are chosen from actual hardware characteristics [20, 21, 45].



(a) 3D Trajectory (b) Position, Y vs. X (c) Position, h vs. X

Figure 3.5: 6-DOF Simulation Trajectory



(a) a_x (b) a_y (c) a_z

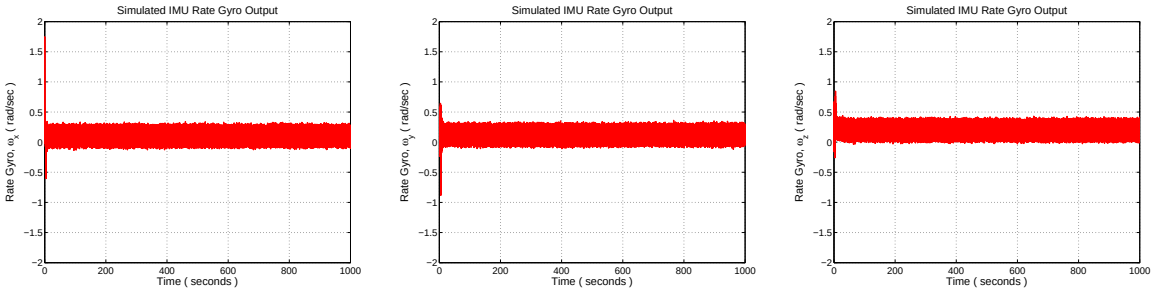
Figure 3.6: Simulated IMU Accelerometer Triad

0, 0) or (q_0, q_1, q_2, q_3) = (1, 0, 0, 0). The initial error covariance matrix is

$$\begin{aligned} P = diag[\ & (5/3\ ft)^2, (5/3)^2, (7/3)^2, \\ & (3/3\ ft/sec)^2, (3/3)^2, (5/3)^2, \\ & (0.01/3)^2, (0.01/3)^2, (0.01/3)^2, (0.01/3)^2, \\ & (0.01/3\ ft/sec^2)^2, (0.01/3)^2, (0.01/3)^2, \\ & (0.01/3\ rad/sec)^2, (0.01/3)^2, (0.01/3)^2\ ]. \end{aligned} \tag{3.85}$$

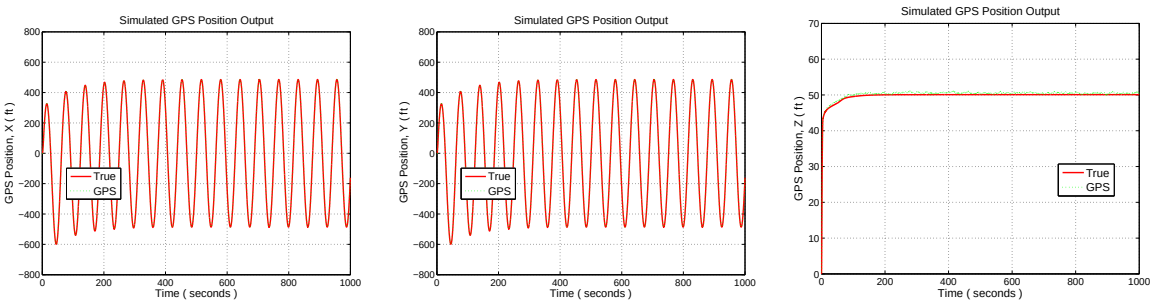
3.5.2 Filter Performance Simulation

In order to assess the performance of a filter by comparing the true state and filtered estimate values, we need both the true and noise-corrupted values. The noise-corrupted value is filtered and compared with the true value, and then the performance of the filter is analyzed by the magnitude of the difference between the true and estimated values. In order to generate a reasonable time history of IMU sensor



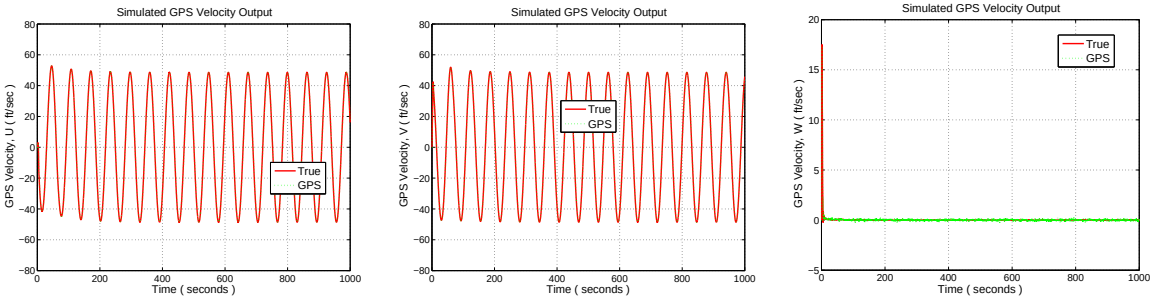
(a) ω_x (b) ω_y (c) ω_z

Figure 3.7: Simulated IMU Rate Gyro Triad



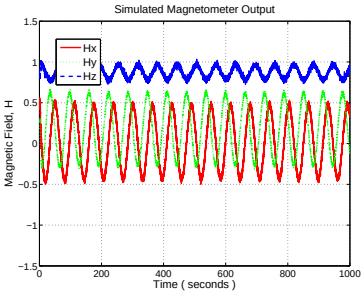
(a) Position, X (b) Position, Y (c) Altitude, h

Figure 3.8: Simulated GPS Position



(a) Velocity, U (b) Velocity, V (c) Velocity, W

Figure 3.9: Simulated GPS Velocity



(a) Normalized H_x

Figure 3.10: Simulated Magnetometer Triad

data, a full 6-DOF simulation is performed for the trajectory motion shown in Figure 3.5. The resulting three-axis body accelerations (or specific forces) and three-axis angular rates generate nominal true IMU sensor outputs. This nominal sensor value is processed to be noise-corrupted with various error sources such as biases, white Gaussian noise, helicopter rotor vibration effects, and so on. The outputs of both the simulated IMU accelerometers and the simulated IMU rate gyros are presented in Figures 3.6 and 3.7, respectively. GPS position data, generated with the first-order lag noise, are shown in Figure 3.8. GPS velocity data, generated with the zero-mean white Gaussian noise model, are shown in Figure 3.9. For comparison purposes, these figures include the true values. The GPS position and velocity data are generated with a latency of 100 $msec$ and 200 $msec$, respectively. Magnetometer sensor data is generated under a zero-mean white Gaussian noise assumption. For the magnetometer model, the World Magnetic Model 2005 (WMM-2005) is used to generate the reference magnetic field in the navigation frame. This reference magnetic field is transformed into the quantity in the body frame and corrupted with zero-mean Gaussian noise as shown in Figure 3.10. The standard deviation of the magnetic field noise is 0.02 in normalized magnetic field quantity. As a measurement variance in the heading angle measurement, $R_{mag} = \sigma_{\psi}^2 = (0.87266\ rad)^2 (\simeq 50\ deg)^2$ is applied to the filter. Figure 3.11 shows the effects of the three-axis magnetometer triad only. The magnetometer triad reduces not only the yaw angle error but also the roll and pitch angle errors.

Using the above artificially-generated IMU, GPS, and magnetometer sensor data set, the performance of the new navigation system is simulated. In order to examine the performance behavior of the newly-developed navigation filter based on the UKF with sequential measurement updates, various filter outputs are presented in Figure 3.12 $\sim$ Figure 3.23. Comparisons of the true trajectory and the navigation

filter output trajectory are shown in Figure 3.12, and the time histories of their errors in Figure 3.13. The $\pm 1\sigma$ values of error covariance histories are also included so that the error time histories can be compared to the error covariance boundaries. Similarly, the velocities of the true and the navigation outputs are compared in Figure 3.14, and the velocity error histories are presented in Figure 3.15. The position and velocity error histories show that the error histories are well-behaved within the $\pm 1\sigma$ error covariance boundary except for during the initial phase of the navigation estimation. Furthermore, the UKF-based navigation system provides better estimation performance by reducing peak values of both position and velocity errors in the initial phase.

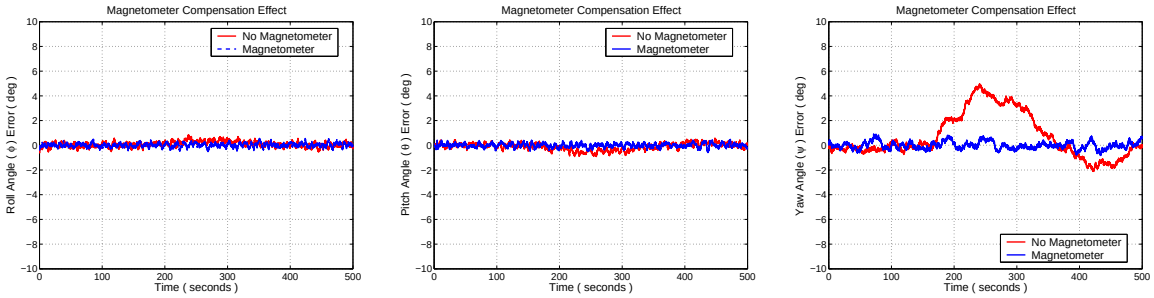
True quaternion and quaternion estimates using the UKF and the EKF navigation systems are compared in Figure 3.16, and the error histories of the quaternion differences are presented in Figure 3.17 with corresponding $\pm 1\sigma$ error covariance boundaries. We see the better error convergence of the UKF-based navigation system than that of the EKF-based navigation system. Comparisons between the true Euler angles and the estimated Euler angles are presented in Figure 3.18 and their corresponding error histories in Figure 3.19. Here, the true Euler angles are obtained from the true quaternion, and the estimated Euler angles are obtained from the quaternion estimates by the UKF-based and the EKF-based navigation systems. The EKF-based estimation results show steady-state nonzero errors in the estimated Euler angles, but the error histories of the UKF-based estimator converge to zero steady-state errors.

Time histories of the three-axis accelerometer biases are given in Figures 3.20(a) $\sim$ 3.20(c). The UKF-based estimator provides better convergence behavior from zero initial conditions to some nonzero bias values. These results are not so surprising considering the accuracy analysis of the EKF-based and the UKF-based methods discussed in Section 2.4.2. Error histories of the accelerometer biases are given in

Figures 3.21(a) $\sim$ 3.21(c), which show better error convergence and covariance behavior of the UKF-based estimator than those of the EKF-based estimator.

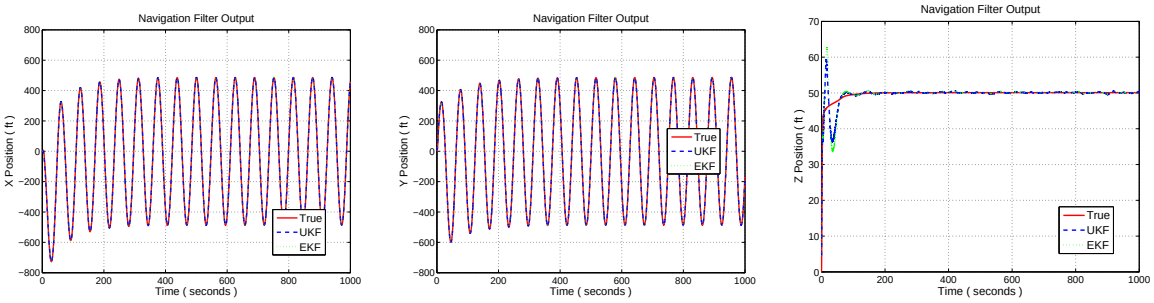
Time histories of the rate gyro biases and their associated error histories are presented in Figures 3.22(a) $\sim$ 3.22(c) and Figures 3.23(a) $\sim$ 3.23(c), respectively. The UKF-based estimator gives better performance in terms of both error time histories and error steady-state convergence.

Figure 3.24(a) compares the time histories of the RMS (root mean squared) position error of the EKF-based navigation system and that of the UKF-based navigation system. The time histories of RMS velocity error using the two navigation systems are also compared in Figure 3.24(b). The position and velocity RMS error histories show that the UKF-based navigation system has better performance than the EKF-based navigation system especially in the initial convergence by reducing transient peaks. Roll, pitch, and yaw angle errors of the EKF-based navigation system are compared with those of the UKF-based navigation system in Figure 3.25(a) $\sim$ Figure 3.25(c). The UKF-based navigation system also exhibits better performance in the convergence to a zero steady-state error compared to that of the EKF-based navigation system. In the RMS error comparison of accelerometer biases presented in Figure 3.26(a), the EKF-based estimator cannot converge to the given nonzero setting of the accelerometer biases, but the UKF-based estimator shows some convergence with only a small steady-state error. In the RMS error comparison of the rate gyro biases given in Figure 3.26(b), both estimators show some convergence in the initial phase of navigation, but the UKF-based estimator provides smaller steady-state error.



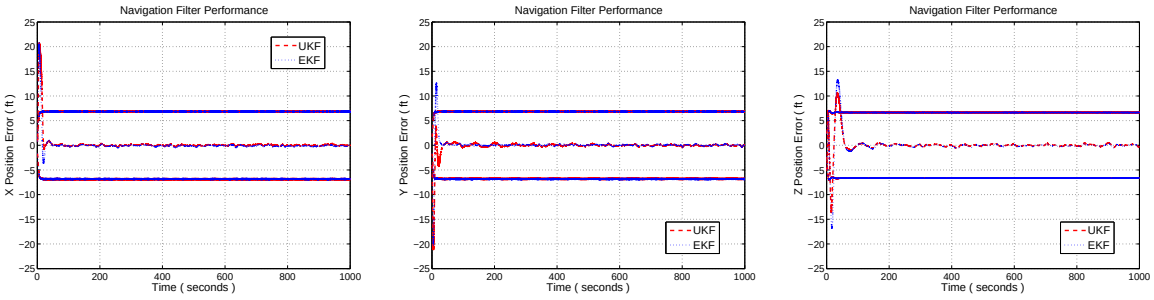
(a) Roll Angle Error (b) Pitch Angle Error (c) Yaw Angle Error

Figure 3.11: Heading Update Effect by Magnetometer Triad



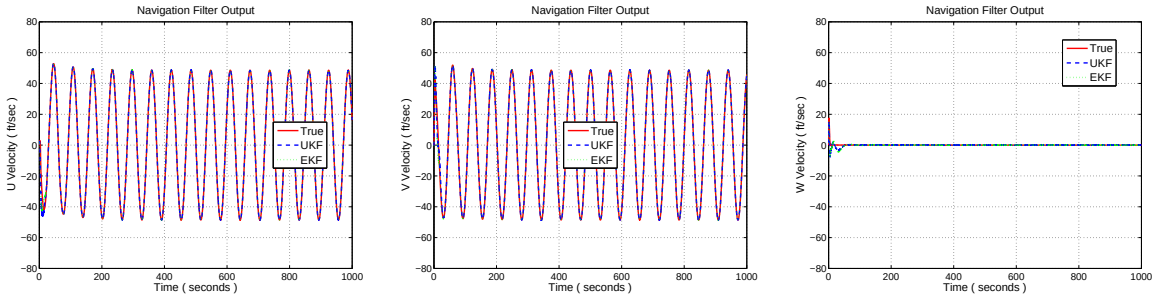
(a) Position, X (b) Position, Y (c) Altitude, h

Figure 3.12: Comparison of True and Navigation Output (Position)



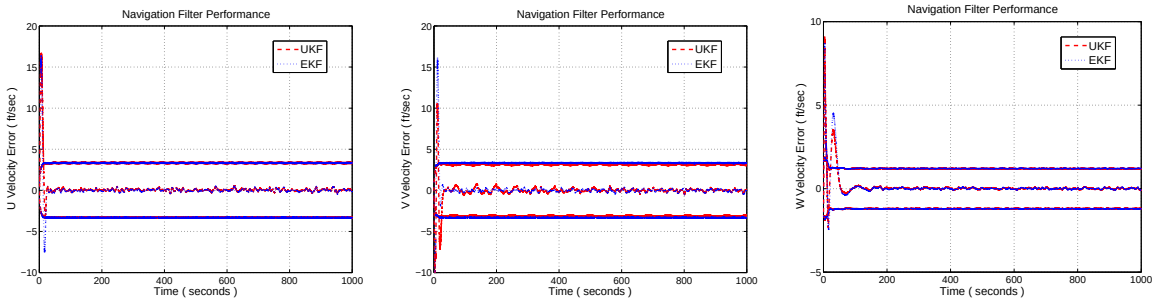
(a) X Position Error (b) Y Position Error (c) Altitude Error

Figure 3.13: Navigation Filter Performance (Position Error)



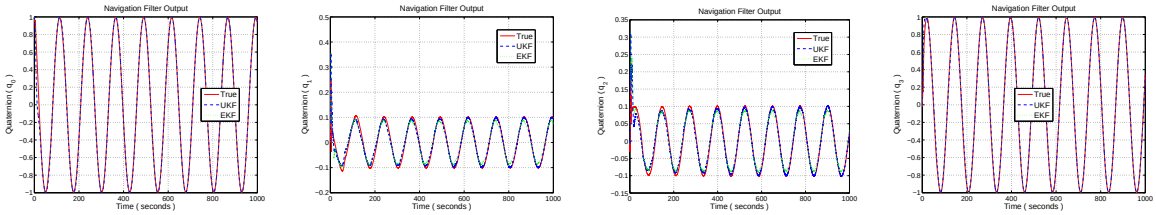
(a) Velocity, U (b) Velocity, V (c) Velocity, W

Figure 3.14: Comparison of True and Navigation Output (Velocity)



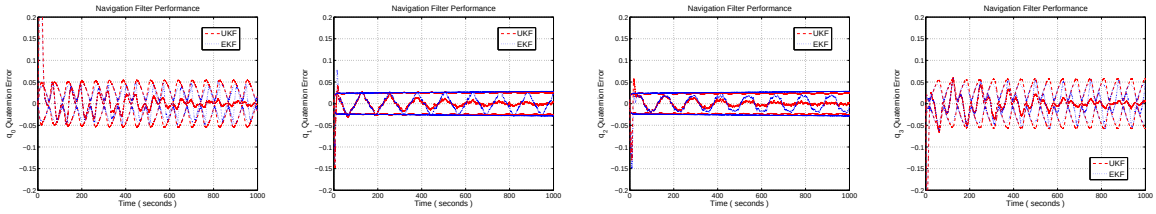
(a) U Velocity Error (b) V Velocity Error (c) W Velocity Error

Figure 3.15: Navigation Filter Performance (Velocity Error)



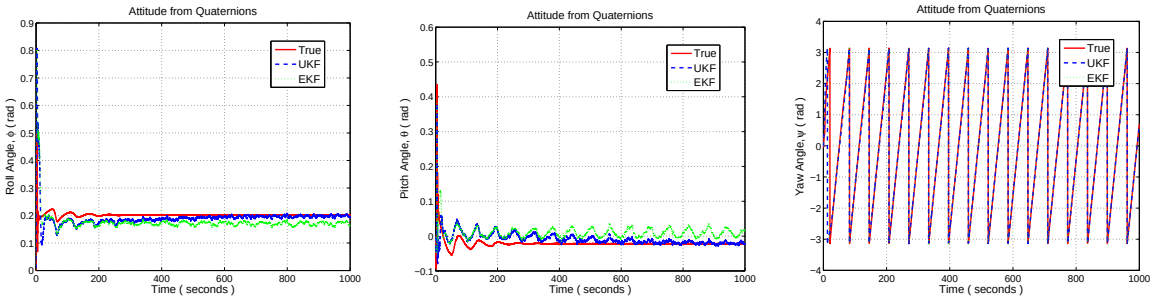
(a) Quaternion, q_0 (b) Quaternion, q_1 (c) Quaternion, q_2 (d) Quaternion, q_3

Figure 3.16: Comparison of True and Navigation Output (Quaternion)



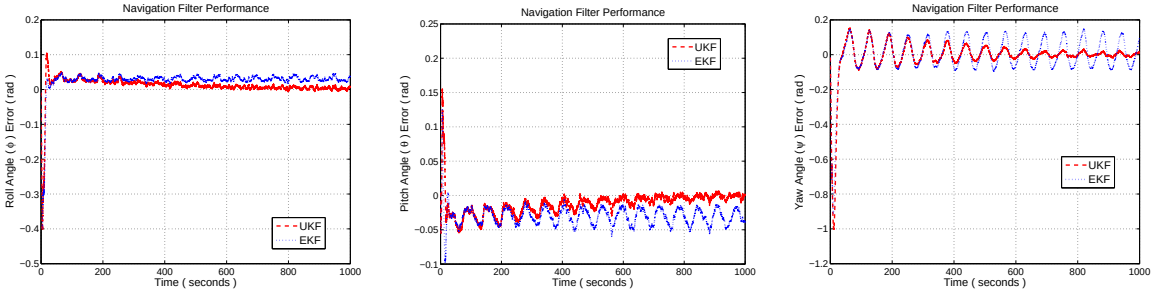
(a) q_0 Quaternion Error (b) q_1 Quaternion Error (c) q_2 Quaternion Error (d) q_3 Quaternion Error

Figure 3.17: Navigation Filter Performance (Quaternion Error)



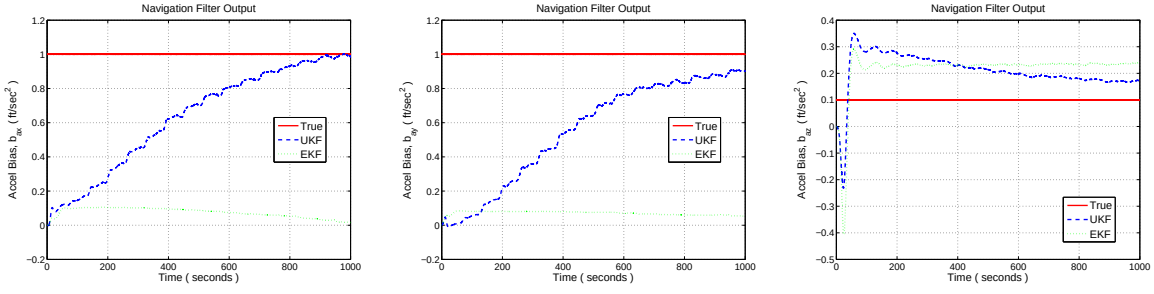
(a) Roll Angle, ϕ (b) Pitch Angle, θ (c) Yaw Angle, ψ

Figure 3.18: Comparison of True and Navigation Output (Euler Angle)



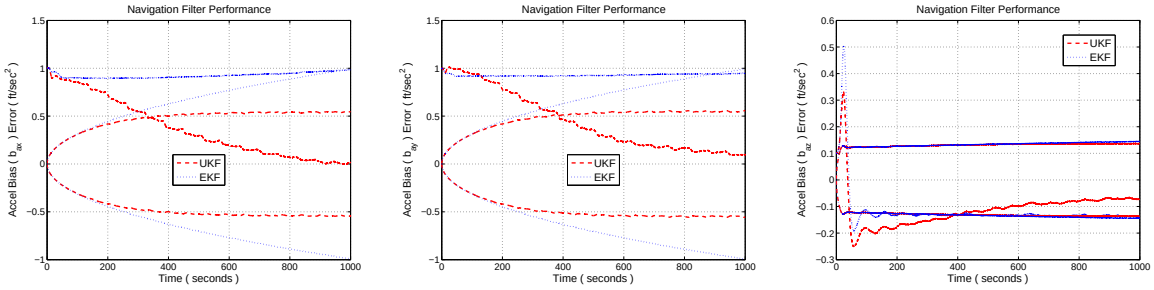
(a) Roll Angle Error (b) Pitch Angle Error (c) Yaw Angle Error

Figure 3.19: Navigation Filter Performance (Euler Angle Error)



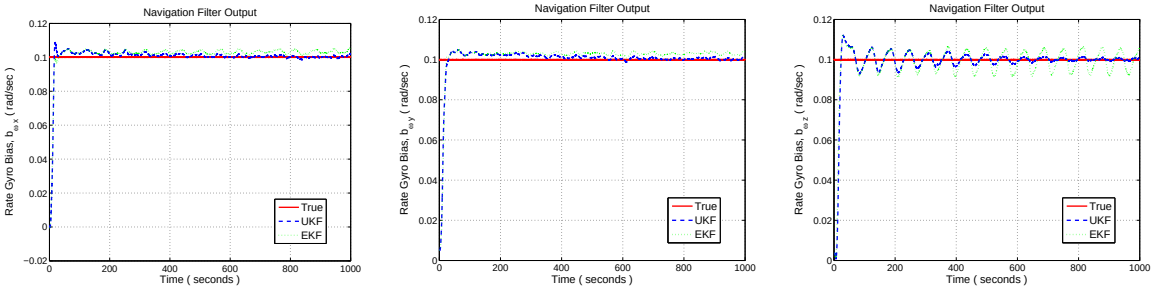
(a) Acceleration Bias, b_{a_x} (b) Acceleration Bias, b_{a_y} (c) Acceleration Bias, b_{a_z}

Figure 3.20: Comparison of True and Navigation Output (Accelerometer Bias)



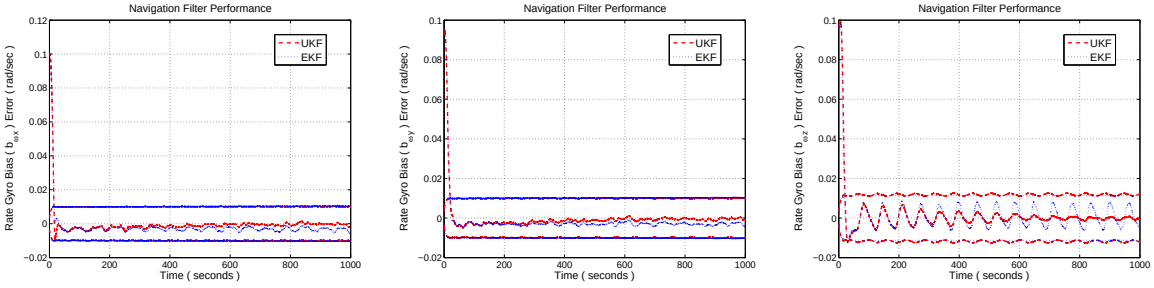
(a) b_{a_x} Acceleration Bias Error (b) b_{a_y} Acceleration Bias Error (c) b_{a_z} Acceleration Bias Error

Figure 3.21: Navigation Filter Performance (Accelerometer Bias Error)



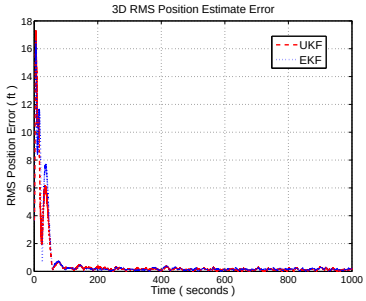
(a) Rate Gyro Bias, b_{ω_x} (b) Rate Gyro Bias, b_{ω_y} (c) Rate Gyro Bias, b_{ω_z}

Figure 3.22: Comparison of True and Navigation Output (Rate Gyro Bias)

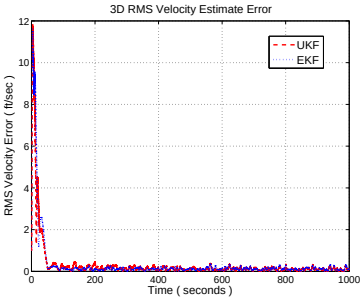


(a) b_{ω_x} Rate Gyro Bias Error (b) b_{ω_y} Rate Gyro Bias Error (c) b_{ω_z} Rate Gyro Bias Error

Figure 3.23: Navigation Filter Performance (Rate Gyro Bias Error)

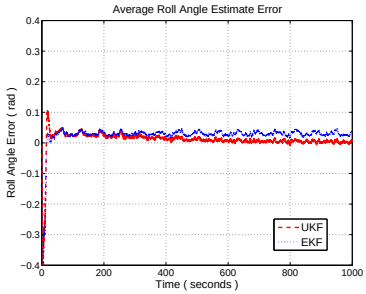


(a) RMS Position Error

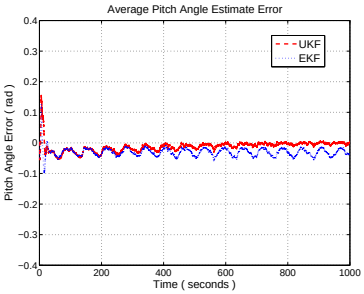


(b) RMS Velocity Error

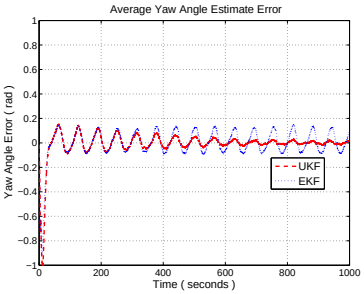
Figure 3.24: Navigation Filter RMS Position and Velocity Error (Comparison of EKF and UKF)



(a) Roll Attitude Error

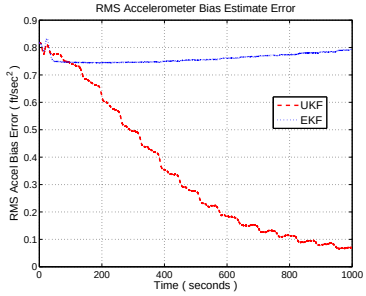


(b) Pitch Attitude Error

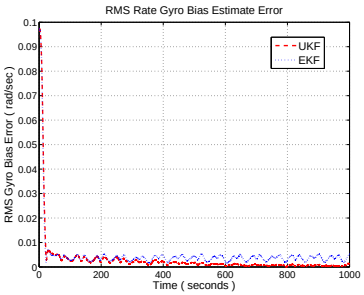


(c) Yaw Attitude Error

Figure 3.25: Navigation Filter Attitude Error (Comparison of EKF and UKF)



(a) RMS Accel Bias Error



(b) RMS Rate Gyro Bias Error

Figure 3.26: Navigation Filter RMS Bias Error (Comparison of EKF and UKF)

3.6 Summary

This chapter discusses the development of an integrated inertial navigation system for use in the autonomous research UAV helicopter. The integrated navigation system is based on low-cost strapdown IMU sensors and several additional aiding sensors such as a DGPS, a three-axis magnetometer, and so on. Including the detailed process and measurement models, the integrated strapdown navigation systems using both the EKF and the UKF are discussed in detail. The resulting implementation of the EKF-based and the UKF-based integrated navigation filters are compared and discussed. Both the EKF- and the UKF-based integrated navigation systems have advantages and disadvantages. The EKF-based navigation system requires less computational power and easily handles the multi-rate sensor fusion problem if we use the sequential measurement update strategy. Due to the necessity of computing messy Jacobian matrices in the linearization of the nonlinear process and measurement models, the EKF-based navigation system requires troublesome work to implement. On the other hand, the UKF-based navigation system obviates the Jacobian matrix computations of the process and measurement models, but in general it requires potentially more computational cost depending on the system characteristics.

In order to effectively solve the multi-rate sensor fusion problem in which a series of aiding sensor data with different measurement vector size and different update-rate are fused with high-rate IMU sensor measurements, a new integrated navigation system algorithm is developed based on the UKF with sequential measurement updates described in Section 2.4.4. The advantages of the UKF with sequential measurement updates are the easy treatment of the multi-rate sensor fusion and sensor latency problems, the easy addition of new aiding sensors to an existing navigation system while retaining the advantages of the standard UKF such as high-order accuracy in nonlinear filtering and no necessity of Jacobian matrix computations for nonlinear process and measurement models. By using the sequential measurement UKF in our

integrated navigation system, we can combine the advantages of both the standard UKF and the sequential measurement strategy with the potential penalty of a slightly increased computational cost.

In terms of accuracy, the UKF-based navigation system shows better estimation performance of all state components than the EKF-based system in the high-fidelity 6-DOF simulation of a horizontal circular trajectory with nonzero accelerometer and rate gyro biases. These findings may also apply to simulations that involve higher nonlinearity in motion such as aggressive vehicle maneuvers.

CHAPTER 4

VISION-BASED TRACKING SYSTEM BASED ON UNSCENTED KALMAN FILTER

4.1 Introduction

An accurate awareness of the surrounding environment using a vision-based tracking system depends on both the quality of the image processing outputs and the performance of the vision-based state estimator that estimates useful environment state information (e.g., a position and a velocity relative to obstacles or other vehicles) by using the image processing outputs as estimator measurement inputs. The details of the image processing algorithms used in our current work are presented in references [43–45], and we limit our attention to the design of the vision-based state estimator. Accuracy in the estimation of surrounding environment including the relative location and velocity of stationary/moving obstacles and friendly/enemy vehicles is crucial for ownship safety and for mission completion in the presence of noisy and uncertain vision-sensor based information about the environment. Research applications in this regards are in formation flight [46, 93, 104, 105], target tracking [82], obstacle avoidance [68, 126], trajectory planning [27], and the generation of advanced guidance and control commands [65, 98, 99, 125, 127].

Research on vision-based relative state estimation with applications to formation flight or target tracking is more specifically related to current vision-based tracking system. Ivey and Johnson [42] explored the square-root UKF to identify location and orientation of stationary objects using a vision sensor and compared the results with those of the EKF. Campbell and Wheeler [9] applied the UKF to a vision-based

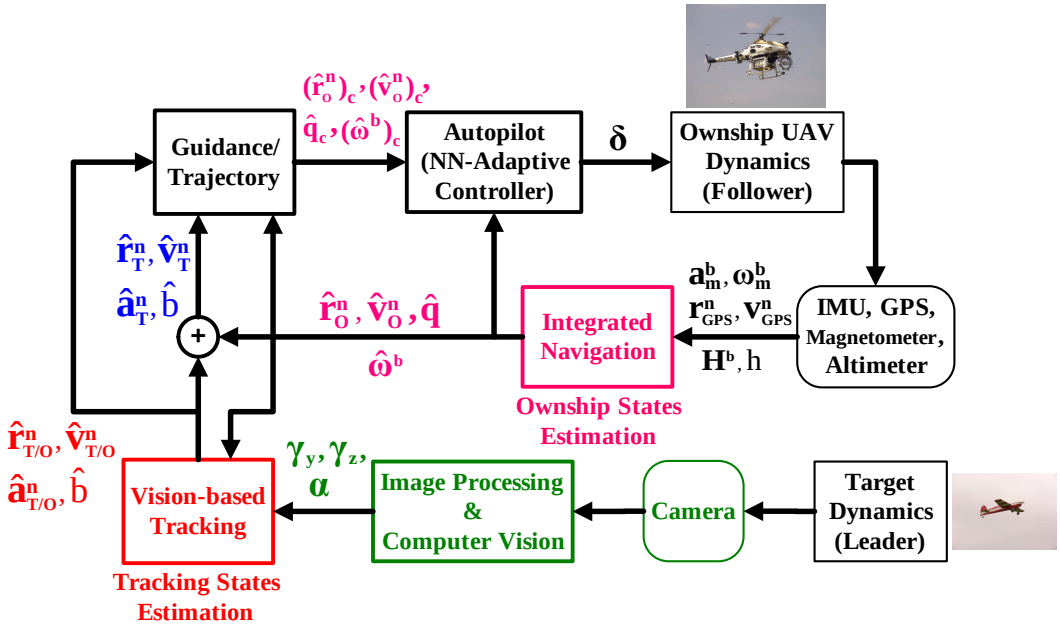


Figure 4.1: Closed-loop UAV Vision Navigation System

geolocation tracking system. Johnson et al. [45, 46] employed the EKF using modified polar (MP) coordinates to design a vision-based state estimator that estimates relative kinematics, a size, and lateral acceleration of a target in the scenario of horizontal formation flight. Current work is an attempt to extend this vision-based state estimator to a more general vision-based air-to-air tracking system that allows target motion to both lateral and longitudinal directions. If we attempt to extend the target model in the framework of the EKF with MP coordinates, the estimator formulation involves very complicated computation of jacobian matrices. In order to circumvent this difficulty without losing accuracy in the highly nonlinear estimation problem, a new state-space estimation model based on Cartesian coordinates is formulated and solved by the UKF. The use of the vision-based tracking system in the closed-loop UAV GNC system of current work is presented in Figure 4.1.

Current vision-based tracking problem is closely related to the typical bearings-only problem [16, 71, 101] which, sometimes referred to as "target motion analysis" (TMA), is the estimation of the relative kinematics such as the relative position and

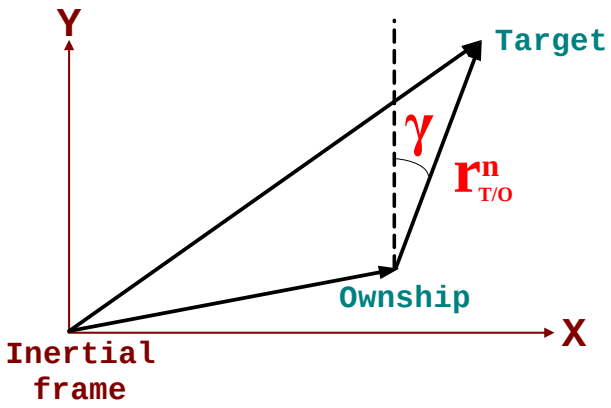


Figure 4.2: Typical Bearings-Only Problem

velocity of a moving target in a planar motion using only noise-corrupted bearing measurements (Figure 4.2). Due to a variety of important practical applications, even the relatively simple bearings-only problem has resulted in numerous research efforts to overcome the difficulty associated with inherent nonlinearity and observability issues. One notable contribution in this field was performed by Aidala and Hammel [1]. They introduced the modified polar (MP) coordinates to overcome the limitations of the EKF and could obtain a filter formulation that is stable and asymptotically unbiased. Some difficulties applying this filter are in that its formulation requires somewhat involved filter initialization and complex derivation of Jacobian matrices since MP coordinates include several indirect physical variables. Among other efforts to circumvent EKF limitations, Grossman [35] suggested an approach to use Hybrid coordinates, and Lerro and Bar-Shalom [71] proposed the bias compensated Cartesian filter. Other recent efforts are to apply nonlinear filters to this bearings-only tracking problem. Xu and Liping [131] applied the UKF to this problem and compared with the results obtained by the EKF. Gordon et al. [33] solved this problem by their historical resampling-based particle filter. For more detailed literature survey on this subject, refer to Section 1.4.1.

The purpose of this research is to develop a vision-based tracking system that

estimates relative kinematics and target characteristics based on vision-only information of a target in the framework of the UKF. This problem is an extended version of the bearings-only problem. We attempt to estimate not only relative kinematics such as the relative position and velocity in three-dimensional space but also the target characteristics such as the target size and target acceleration components using two bearing angles and a subtended angle (Figure 4.6).

The images of the target aircraft projected on the video-camera image plane of the ownship aircraft are captured and processed into the vision information. The vision information obtained by the application of a geometric active contour method on the raw images captured by the onboard camera, consists of target wing-tip and center locations on the onboard camera image plane [45]. The target center point is used to derive the azimuth and elevation angles of the target in the camera frame, and the two wing-tip points provide target subtended angle information that ultimately provides the target size. Vision information for the target tracking system in this work is composed of three target angles: an azimuth angle, an elevation angle, and a subtended angle.

In current work, the performance of the new vision-based tracking system using the UKF is simulated in the scenario of formation flight in which an ownship UAV (or follower) maintains some distance from a target aircraft (or leader) based on vision-only information of the target. The estimated states in the ownship UAV are relative three-dimensional positions and velocities, a target size, and target acceleration components. We assume the ownship UAV and the target aircraft have no communication link [105]. This generalization for no communication link makes the current work applicable to a variety of situations such as tracking or avoiding an adversarial air-target, flying in formation with communication loss when using the jamming- or spoofing-vulnerable GPS signals, and performing a silent, cooperative

(a) GTMax Helicopter UAV (b) GTEdge Fixed-Wing UAV

Figure 4.3: GTMax and GTEdge UAV

mission in an adversarial environment without being noticed by enemies. The developed algorithm is to be implemented on GTMax helicopter UAV [20,21,48,49] shown in Figure 4.3(a). A fixed-wing UAV, named GTEdge and shown in Figure 4.3(b), is used as the target aircraft whose image sequence on the image plane of the ownship aircraft (GTMax) provides the real-time image information during the flight test.

The chapter is organized as follows. Section 4.2 describes the overall system for vision-based relative navigation including the details of a process model, a measurement model, and measurements from an image processor. Section 4.3 presents the filter initializations of a state vector and a covariance matrix and the simulation results starting these initial conditions. Section 4.4 summarizes the chapter.

4.2 System Description for Vision-Based Relative Navigation

This section describes the formulation details of a vision-based tracking system (or vision-based relative motion estimator) using an unscented Kalman filter (UKF). The state vector is composed of the relative position, the relative velocity, the size, and the lateral and longitudinal acceleration components of the target. Three target angles–the azimuth angle, the elevation angle, and the subtended angle–are used as

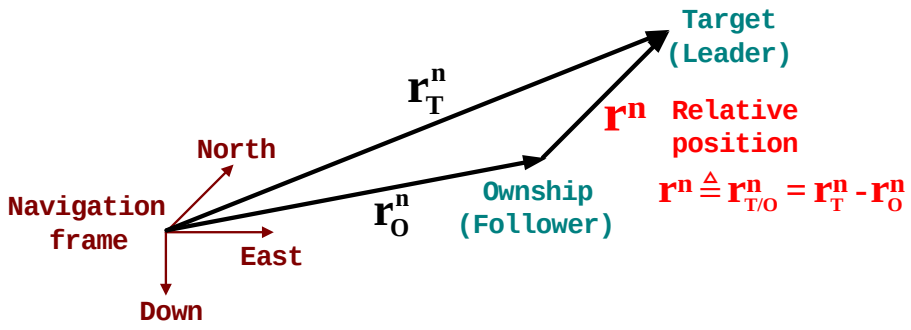


Figure 4.4: Target Relative Position in a Navigation Frame

the measurements obtained from the onboard image processor. By using the UKF, we are able to remove the rather messy step of calculating Jacobian matrices of the nonlinear process and measurement models.

4.2.1 Process Model of Vision-based Relative Navigation

The state vector of the process model for the vision-based relative navigation includes three relative position components, three relative velocity components, a target size, and target lateral and longitudinal acceleration components:

$$\mathbf{x} = \begin{bmatrix} x_{1:3} \\ x_{4:6} \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} \mathbf{r}^n \\ \mathbf{v}^n \\ b \\ a_{lat} \\ a_{long} \end{bmatrix}, \quad \text{where } \mathbf{r}^n = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \ \mathbf{v}^n = \begin{bmatrix} U \\ V \\ W \end{bmatrix}. \tag{4.1}$$

Here, as represented in Figure 4.4, $\mathbf{r}^n$ and $\mathbf{v}^n$ are the relative position vector and the relative velocity vector from an ownship vehicle to a target vehicle in a navigation frame, respectively, b is the target size, a_{lat} is the target lateral acceleration, and a_{long} is the target longitudinal acceleration.

The continuous-time process model can be expressed by these state vector components. Since $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \mathbf{r}^n = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$, we have $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \dot{\mathbf{r}}^n = \mathbf{v}^n = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \triangleq \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$. Similarly, using the fact that $\begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} = \dot{\mathbf{r}}^n = \mathbf{v}^n = \begin{bmatrix} U \\ V \\ W \end{bmatrix}$, we have $\begin{bmatrix} \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \ddot{\mathbf{r}}^n = \dot{\mathbf{v}}^n = \mathbf{a}^n = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} \triangleq \begin{bmatrix} f_4 \\ f_5 \\ f_6 \end{bmatrix}$.

Furthermore, $\dot{x}_7 = \dot{b} = 0 \triangleq f_7$, $\dot{x}_8 = \dot{a}_{lat} = -\frac{1}{\tau_{lat}} a_{lat} = -\frac{1}{\tau_{lat}} x_8 \triangleq f_8$, and $\dot{x}_9 = \dot{a}_{long} == -\frac{1}{\tau_{long}} a_{long} = -\frac{1}{\tau_{long}} x_9 \triangleq f_9$. We assume the random walk model for the target size and the first-order Markov model for target lateral and longitudinal acceleration. τ_{lat} and τ_{long} are lateral and longitudinal maneuver time constants, respectively. We need to express the relative acceleration, $\mathbf{a}^n$, as a function of state variables. We assume here that the target maintain a constant speed, thus target acceleration commands are perpendicular to its total velocity vector[1]. Based on the assumption, target longitudinal acceleration is perpendicular to the target total velocity and lies in the vertical plane generated by the total velocity vector and the vertical direction vector (W_T direction), and target lateral acceleration is also perpendicular to the target total velocity but lies in the horizontal plane (shown in Figure 4.5). Now, we can express the target acceleration as a function of target velocity components and target acceleration commands as follows [110]:

[1]We may consider the case in which a target is in its maximum speed (i.e., constant speed), and its evasive maneuvers are performed by the combination of its longitudinal and lateral acceleration (or pitch and yaw maneuvering motion). This assumption is not so restrictive but used even in missile-target 3-D engagement scenario [110]. This assumption is well suited to a scenario of formation flight that allows target pitch and yaw motion.

$$\left(\mathbf{a}_{\mathbf{T}}^{\mathbf{n}}\right)_x = (a_{long}\sin\theta_2)\cos\theta_1 - a_{lat}\sin\theta_1$$

$$\left(\mathbf{a}_{\mathbf{T}}^{\mathbf{n}}\right)_y = (a_{long}\sin\theta_2)\sin\theta_1 + a_{lat}\cos\theta_1$$

$$\left(\mathbf{a}_{\mathbf{T}}^{\mathbf{n}}\right)_z = -a_{long}\cos\theta_2$$

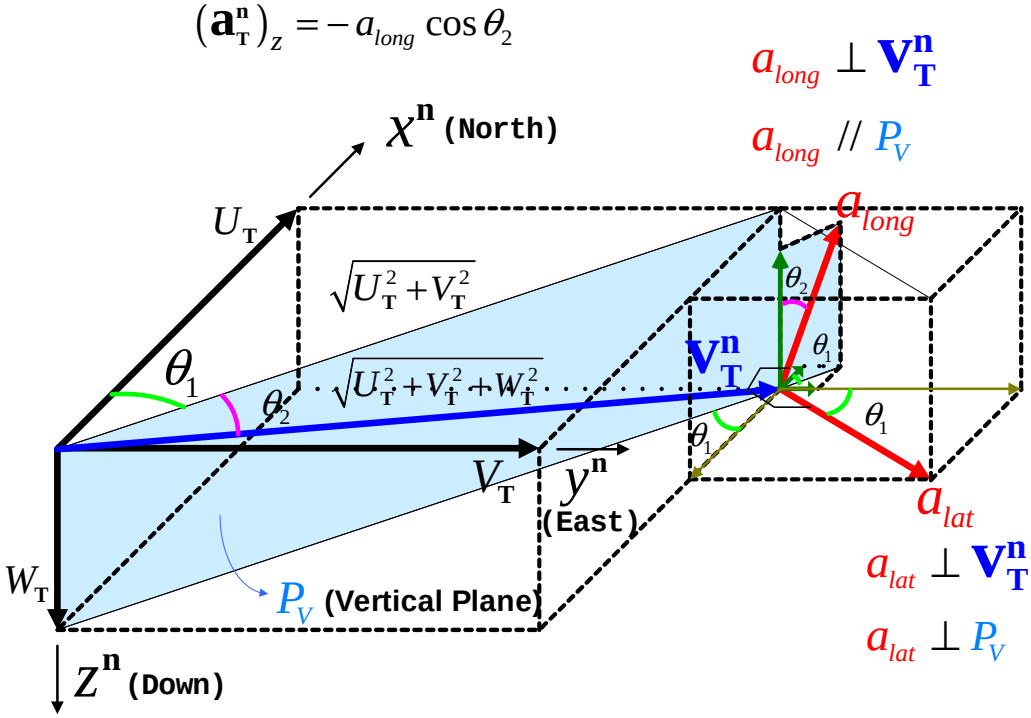


Figure 4.5: Target Acceleration Model

$$\mathbf{a}_T^n = \frac{1}{\sqrt{U_T^2 + V_T^2}} \begin{bmatrix} \frac{U_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & -V_T \\ \frac{V_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & U_T \\ -\frac{U_T^2+V_T^2}{\sqrt{U_T^2+V_T^2+W_T^2}} & 0 \end{bmatrix} \begin{bmatrix} a_{long} \\ a_{lat} \end{bmatrix}, \tag{4.2}$$

where

$$\mathbf{v}_T^n = \begin{bmatrix} U_T \\ V_T \\ W_T \end{bmatrix} = \mathbf{v}^n + \mathbf{v}_O^n = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} + \mathbf{v}_O^n. \tag{4.3}$$

$\mathbf{a}_T^n$ is target acceleration in the navigation frame, $\mathbf{v}_T^n$ is target velocity in the navigation frame, $\mathbf{v}_O^n$ is ownship velocity in the navigation frame, and $\mathbf{v}^n$ is relative velocity vector from the ownship vehicle to the target vehicle in the navigation frame. $\mathbf{v}_F^n$ is assumed to be known from the ownship navigation system. Using these facts, we can express relative acceleration as a function of state variable components.

$$\begin{bmatrix} f_4 \\ f_5 \\ f_6 \end{bmatrix} \triangleq \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \mathbf{a}^n = \mathbf{a}_T^n - \mathbf{a}_O^n = \frac{1}{\sqrt{U_T^2 + V_T^2}} \begin{bmatrix} \frac{U_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & -V_T \\ \frac{V_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & U_T \\ -\frac{U_T^2+V_T^2}{\sqrt{U_T^2+V_T^2+W_T^2}} & 0 \end{bmatrix} \begin{bmatrix} a_{long} \\ a_{lat} \end{bmatrix} - \mathbf{a}_O^n, \tag{4.4}$$

where the ownship acceleration in the navigation frame, $\mathbf{a}_O^n$, is assumed to be known from the ownship navigation system.

Therefore, the process model for the vision-based relative navigation in continuous-time state-space matrix form can be expressed as follows:

$$\dot{\mathbf{x}}(\mathbf{t}) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{w}(t) \tag{4.5}$$

or

$$\begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{b} \\ \dot{a}_{lat} \\ \dot{a}_{long} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ \mathbf{a}^n \\ 0 \\ -\frac{1}{\tau_{lat}} a_{lat} \\ -\frac{1}{\tau_{long}} a_{long} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_v \\ \mathbf{n}_a \\ n_b \\ n_{alat} \\ n_{along} \end{bmatrix}, \tag{4.6}$$

where

$$\mathbf{v}^n = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix}, \quad \mathbf{a}^n = \mathbf{a}_T^n - \mathbf{a}_O^n = \frac{1}{\sqrt{U_T^2 + V_T^2}} \begin{bmatrix} \frac{U_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & -V_T \\ \frac{V_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & U_T \\ -\frac{U_T^2+V_T^2}{\sqrt{U_T^2+V_T^2+W_T^2}} & 0 \end{bmatrix} \begin{bmatrix} x_9 \\ x_8 \end{bmatrix} - \mathbf{a}_O^n, \tag{4.7}$$

$$\begin{bmatrix} U_T \\ V_T \\ W_T \end{bmatrix} = \mathbf{v}_T^n = \mathbf{v}^n + \mathbf{v}_O^n = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} + \mathbf{v}_O^n. \tag{4.8}$$

In a matrix component form, we have

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \frac{1}{\sqrt{U_T^2+V_T^2}} \begin{bmatrix} \frac{U_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & -V_T \\ \frac{V_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & U_T \\ -\frac{U_T^2+V_T^2}{\sqrt{U_T^2+V_T^2+W_T^2}} & 0 \end{bmatrix} \begin{bmatrix} x_9 \\ x_8 \end{bmatrix} - \mathbf{a}_O^n \\ 0 \\ -\frac{1}{\tau_{lat}} x_8 \\ -\frac{1}{\tau_{long}} x_9 \end{bmatrix} + \begin{bmatrix} \mathbf{n}_v \\ \mathbf{n}_a \\ n_b \\ n_{alat} \\ n_{along} \end{bmatrix}, \tag{4.9}$$

where

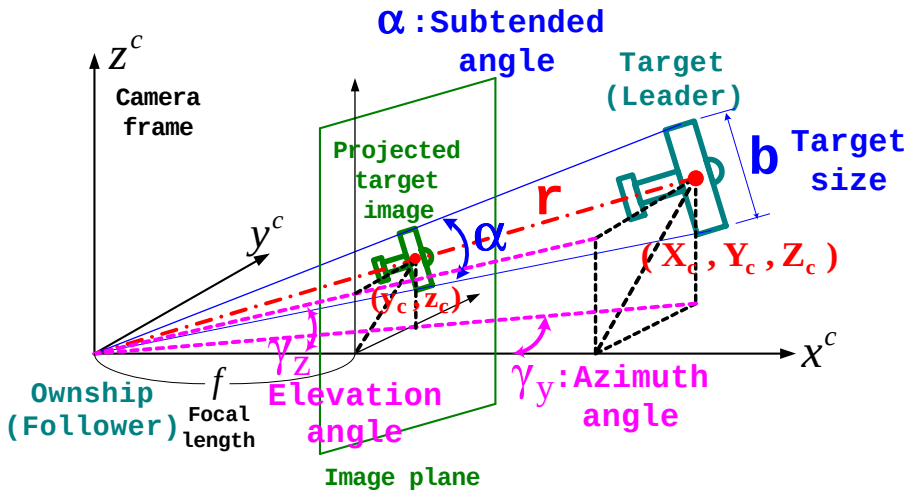


Figure 4.6: Projected Target Image on the Image Plane of a Pinhole Camera

$$\begin{bmatrix} U_T \\ V_T \\ W_T \end{bmatrix} = \mathbf{v}_T^n = \mathbf{v}^n + \mathbf{v}_O^n = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} + \mathbf{v}_O^n. \tag{4.10}$$

The ownship velocity $\mathbf{v}_O^n$ and ownship acceleration $\mathbf{a}_O^n$ in the navigation frame are assumed to be known from the ownship navigation system.

4.2.2 Measurement Model of Vision-based Relative Navigation

For the measurement quantities, we use three target angles: an azimuth angle γ_y, an elevation angle γ_z, and a subtended angle α [45, 68], illustrated in Figure 4.6. These three angles have the following relationships:

$$\tan\gamma_y = \frac{Y_c}{X_c}, \tag{4.11}$$

$$\tan\gamma_z = \frac{Z_c}{X_c}, \tag{4.12}$$

$$\tan\frac{\alpha}{2} = \frac{b}{2\,r} = \frac{b}{2\sqrt{X_c^2 + Y_c^2 + Z_c^2}}, \tag{4.13}$$

where $\mathbf{r}^c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = C_n^c \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = C_n^c \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ is the target relative position in a camera frame, C_n^c is the transformation matrix from the navigation frame to the camera frame, which is given by the ownship navigation system, and $r = \sqrt{X_c^2 + Y_c^2 + Z_c^2}$ is the relative distance between the ownship aircraft and the target aircraft. Now, we can express the three measurement angles as a function of state variables as follows:

$$\mathbf{y} = \begin{bmatrix} \gamma_y \\ \gamma_z \\ \alpha \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{Y_c}{X_c}\right) \\ \tan^{-1}\left(\frac{Z_c}{X_c}\right) \\ 2\ \tan^{-1}\left(\frac{x_7}{2\sqrt{X_c^2+Y_c^2+Z_c^2}}\right) \end{bmatrix}, \tag{4.14}$$

where $\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{r}^c = C_n^c\ \mathbf{r}^n = C_n^c \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, and C_n^c is assumed to be known from the ownship navigation system.

4.2.3 Measurement from the Image Processor

From the image processor, we receive three target data points in the camera image plane: the center point (y_c, z_c), the left wingtip point (y_l, z_l), and the right wingtip point (y_r, z_r).

Using the characteristics of a pinhole camera, we have the following relationships for the target azimuth angle and the target elevation angle:

$$\tan\gamma_y = \frac{y_c}{f} = \frac{Y_c}{X_c}, \tag{4.15}$$

$$\tan\gamma_z = \frac{z_c}{f} = \frac{Z_c}{X_c}, \tag{4.16}$$

where f is the focal length of a pinhole camera, (y_c, z_c) is the projected target center position on the camera image plane, and $\mathbf{r}^c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$ is the relative target position in

the camera frame. Since the relative distance from the origin of the camera frame to the center point of the projected target image on the image plane is $\sqrt{f^2 + y_c^2 + z_c^2}$, and the projected target size on the image plane is $\sqrt{(y_l - y_r)^2 + (z_l - z_r)^2}$, the target subtended angle can be expressed as follows:

$$\tan\frac{\alpha}{2} = \frac{\sqrt{(y_l - y_r)^2 + (z_l - z_r)^2}}{2\sqrt{f^2 + y_c^2 + z_c^2}} = \frac{b}{2\,r} = \frac{b}{2\sqrt{X_c^2 + Y_c^2 + Z_c^2}}. \tag{4.17}$$

Now, the three angle measurements from the image processor are obtained as follows:

$$\mathbf{y} = \begin{bmatrix} \gamma_y \\ \gamma_z \\ \alpha \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{y_c}{f}\right) \\ \tan^{-1}\left(\frac{z_c}{f}\right) \\ 2\ \tan^{-1}\left(\frac{\sqrt{(\frac{y_l}{f} - \frac{y_r}{f})^2 + (\frac{z_l}{f} - \frac{z_r}{f})^2}}{2\sqrt{1 + \left(\frac{y_c}{f}\right)^2 + \left(\frac{z_c}{f}\right)^2}}\right) \end{bmatrix}. \tag{4.18}$$

4.3 Simulation and Results

Standard UKF described in section 2.4.3 is applied to this simulation.

4.3.1 Initialization and Noise Covariance Setting

Filter initialization is performed by using the current measurement.

1) Initialization of the State Vector

$$\begin{bmatrix} x_1(t_0) \\ x_2(t_0) \\ x_3(t_0) \end{bmatrix} = \begin{bmatrix} X(t_0) \\ Y(t_0) \\ Z(t_0) \end{bmatrix} = C_c^n \begin{bmatrix} X_c(t_0) \\ Y_c(t_0) \\ Z_c(t_0) \end{bmatrix}, \tag{4.19}$$

$$\begin{bmatrix} x_4(t_0) \\ x_5(t_0) \\ x_6(t_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.20}$$

$$x_7(t_0) = \hat{b}_0, \tag{4.21}$$

$$x_8(t_0) = 0.0, \tag{4.22}$$

$$x_9(t_0) = 0.0, \tag{4.23}$$

where

$$X_c = \frac{\hat{b}_0}{2 \tan\frac{\alpha}{2}\sqrt{1+\tan^2\gamma_y+\tan^2\gamma_z}}, \tag{4.24}$$

$$Y_c = X_c \tan\gamma_y, \tag{4.25}$$

$$Z_c = X_c \tan\gamma_z. \tag{4.26}$$

($\hat{b}_0$=8.75 ft: estimate of target wingspan or target size)

2) Initialization of the Error Covariance Matrix

$$P(t_0) = diag\{\sigma_X^2, \sigma_Y^2, \sigma_Z^2, \sigma_U^2, \sigma_V^2, \sigma_W^2, \sigma_b^2, \sigma_{alat}^2, \sigma_{along}^2\}, \tag{4.27}$$

where $\sigma_X = \sigma_Y = \sigma_Z = 1\ ft$, $\sigma_U = \sigma_V = \sigma_W = 1\ ft/sec$, $\sigma_b = 0.02 * \hat{b}_0\ ft$, $\sigma_{alat} = 10\ ft/sec^2$, $\sigma_{along} = 10\ ft/sec^2$.

3) Process Noise

$$Q_k = \Delta t\ Q, \tag{4.28}$$

$$Q = diag\{{\sigma_{\dot{X}}}^2, {\sigma_{\dot{Y}}}^2, {\sigma_{\dot{Z}}}^2, {\sigma_{\dot{U}}}^2, {\sigma_{\dot{V}}}^2, {\sigma_{\dot{W}}}^2, \sigma_{\dot{b}}^2, {\sigma_{\dot{a}_{lat}}}^2, {\sigma_{\dot{a}_{long}}}^2\}, \tag{4.29}$$

where $\sigma_{\dot{X}} = \sigma_{\dot{Y}} = \sigma_{\dot{Z}} = 5\ ft/sec$, $\sigma_{\dot{U}} = \sigma_{\dot{V}} = \sigma_{\dot{W}} = 10\ ft/sec^2$, $\sigma_{\dot{b}} = 0.01 * \hat{b}_0\ ft/sec$, $\sigma_{\dot{a}_{lat}} = 2\ ft/sec^3$, $\sigma_{\dot{a}_{long}} = 2\ ft/sec^3$.

4) Measurement Noise

$$R = diag\{\sigma_{\gamma_y}^2, \sigma_{\gamma_z}^2, \sigma_\alpha^2\}, \tag{4.30}$$

where $\sigma_{\gamma_y} = 5\ deg$, $\sigma_{\gamma_z} = 4\ deg$, $\sigma_\alpha = 3\ deg$.

5) Markov model parameters

$\tau_{lat} = \tau_{long} = 60\ sec$ for a lazy turn [77].

4.3.2 Simulation Results

This chapter describes a vision-based relative motion estimator (or vision-based tracking system) in the formation flight of two UAV's. We consider the state estimation of

(a) GTMax and GTEdge in Formation Flight (b) GTEdge Image on GTMax Camera Plane

Figure 4.7: GTMax and GTEdge in Formation Flight

relative motion in the scenario in which an ownship UAV (GTMax) maintains some distance from a target aircraft (GTEdge) based on vision-only information about the target (Figure 4.7(a)). The navigation of an ownship aircraft relative to a target aircraft is performed by vision-only information from a single camera fixed to the ownship aircraft. The images of the target aircraft projected on the video-camera plane of the ownship aircraft are captured and processed into vision information. One typical image of the target aircraft captured by the ownship aircraft is presented in Figure 4.7(b). The image measurement produces the target relative position and velocity, the target size, and the target acceleration components. In order to evaluate the performance of the vision-based tracking filter, vision information obtained through a real-time flight test is post-processed by the filter. The real-time vision information, obtained by a geometric active contour method on an onboard computer, is composed of three points of the target aircraft on the camera image plane. These target points are the center point, the left wingtip point, and the right wingtip point in the camera image plane in Figures 4.8(a) and 4.8(b). Figure 4.8(c) shows the image registration status in which "1" denotes an image registered (success) and "0" denotes an image not registered (failure). The target center point on the camera image plane provides information about the azimuth and the elevation angles of the

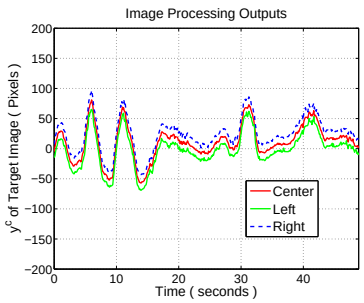
target in the camera frame, and the two wingtip points provide information about the subtended angle of the target that provides the target size. In order to compare the UKF-based tracking system with the EKF-based system, estimation results obtained by the EKF-based state estimator given in [46] are also presented. Figure 4.9 represents four components of EKF measurements, three components of the unit vector that represents the direction from ownship to target center and the subtended angle. Using the EKF-based tracking system with vision-only measurements, relative range is tightly regulated by the closed-loop guidance of the ownship aircraft in the scenario of formation flight. The image processing outputs obtained through this flight test are used to compute the UKF measurements. Figure 4.10 represents three target angle measurements–the azimuth angle, the elevation angle, and the subtended angle–obtained by image processing outputs and used as the inputs to the UKF-based system. Since the relative range is tightly regulated in the flight test, the subtended angle measurement is nearly constant in Figure 4.10(c). Using the vision information measurement, ownship aircraft estimates the relative position, the relative velocity, the size, and the acceleration components of the target in the framework of an unscented Kalman filter (UKF). The UKF is applied to the relative motion estimator due to the highly nonlinear characteristics of the problem at hand.

The vision-based estimation results about the target-ownship relative motion and target characteristics are compared to actual data that are independently obtained from the onboard integrated navigation systems of both aircraft during a flight test. The relative position, the target position relative to the ownship position, in the navigation frame is presented in Figure 4.11. The relative position components of the vision-based estimator outputs are compared with the actual flight test results. The actual relative position components from the flight test are computed from the recorded data by the onboard navigation systems (mainly from the GPS/INS) of each aircraft. The actual positions of each aircraft are independently recorded during

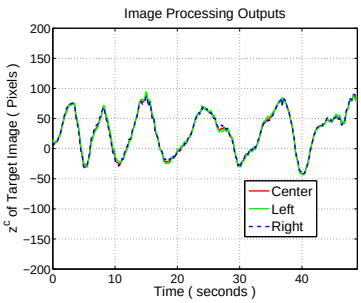
the flight test for comparison purposes. Similarly, relative velocity components and relative acceleration components of estimator outputs are compared with the flight test results, shown in Figures 4.12 and 4.13. The amount of the absolute error between the estimator outputs and flight test results for the relative position, the relative velocity, and the relative acceleration components are presented in Figures 4.14-4.16. Comparisons show that the UKF-based tracking system provides better estimation performance especially in the initial phase.

Figure 4.17 compares the target characteristics of the true and the estimated outputs in terms of the size, the lateral acceleration, and the longitudinal acceleration of the target. Figure 4.17(b) shows the rapid convergence of the UKF-based system compared to the EKF-based system in the estimation of the lateral acceleration from 0 to around 8 ft/sec^2. This better estimation of target acceleration reduces three-components of target relative position errors in the initial phase shown in Figure 4.11. Since the flight test was performed in nearly planar circular motion, the target lateral acceleration is nearly constant as a centripetal acceleration form, and the target longitudinal acceleration is approximately zero. If the target moves in the vertical plane, the UKF-based system will get much better estimation results than the EKF-based system since the EKF-based system does not include a target acceleration model in the vertical direction.

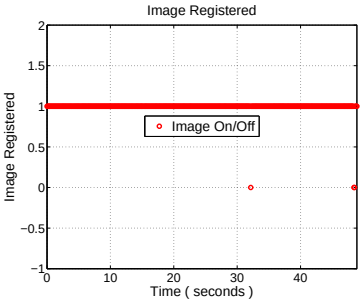
Comparisons indicate that the vision-based estimation filter provides satisfactory estimation results and thus successfully overcomes the highly nonlinear system characteristics in the UKF framework.



(a) Target on y^c Image Frame (pixels)

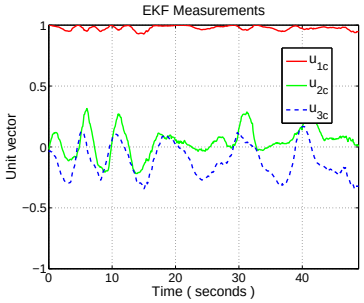


(b) Target on z^c Image Frame (pixels)

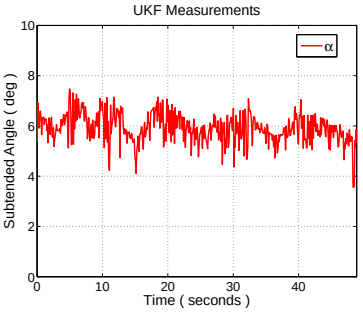


(c) Image Registered

Figure 4.8: Image Processing Outputs of the Target Position

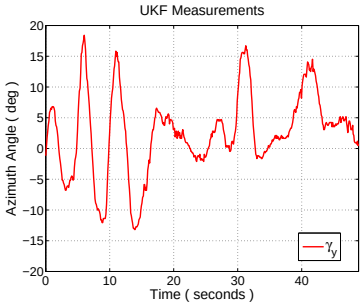


(a) Unit Vector Components, $\mathbf{u}_c$

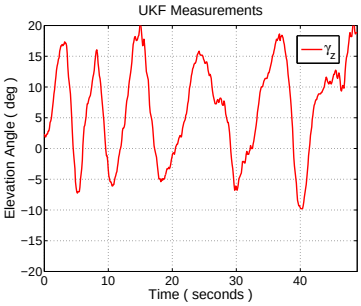


(b) Subtended Angle, α (degrees)

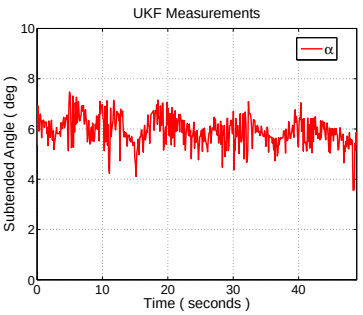
Figure 4.9: EKF Measurements



(a) Azimuth Angle, γ_y (degrees)

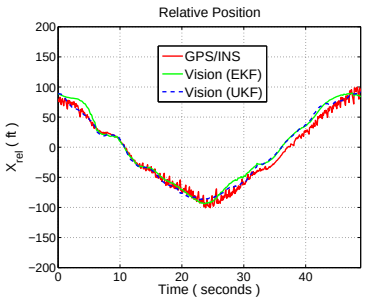


(b) Elevation Angle, γ_z (degrees)

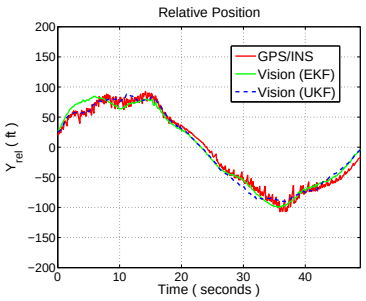


(c) Subtended Angle, α (degrees)

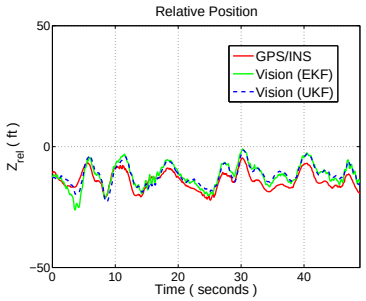
Figure 4.10: UKF Measurements



(a) Relative Position, X (ft)

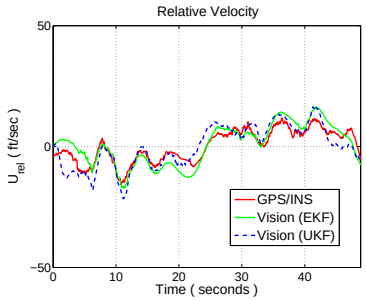


(b) Relative Position, Y (ft)

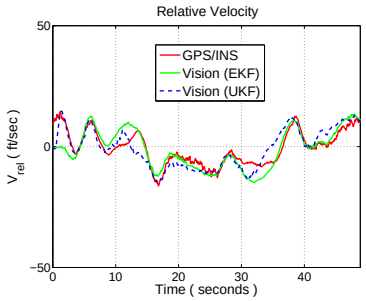


(c) Relative Position, Z (ft)

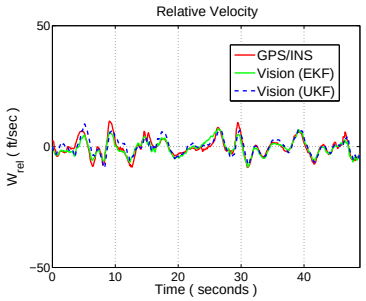
Figure 4.11: Relative Position



(a) Relative Velocity, U (ft/sec)

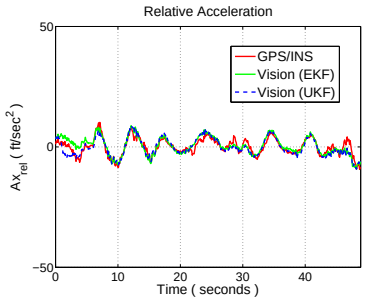


(b) Relative Velocity, V (ft/sec)

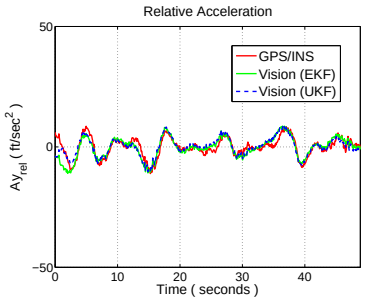


(c) Relative Velocity, W (ft/sec)

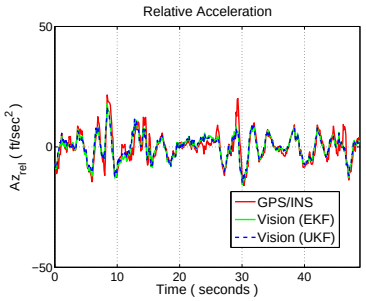
Figure 4.12: Relative Velocity



(a) Relative Acceleration, A_x (ft/sec^2)

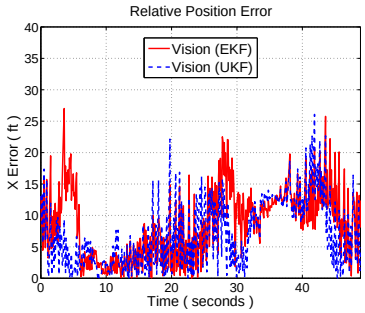


(b) Relative Acceleration, A_y (ft/sec^2)

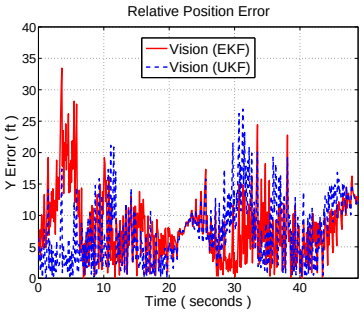


(c) Relative Acceleration, A_z (ft/sec^2)

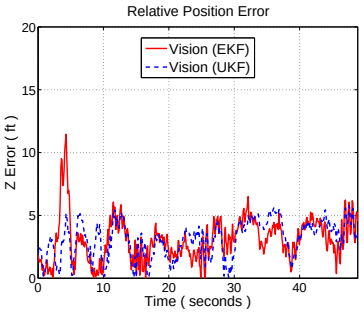
Figure 4.13: Relative Acceleration



(a) X Relative Position Error (ft)

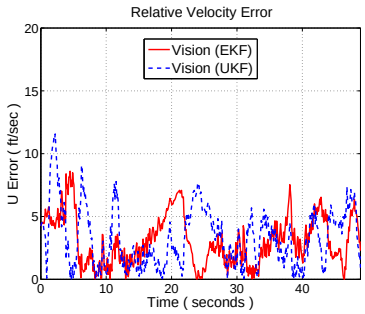


(b) Y Relative Position Error (ft)

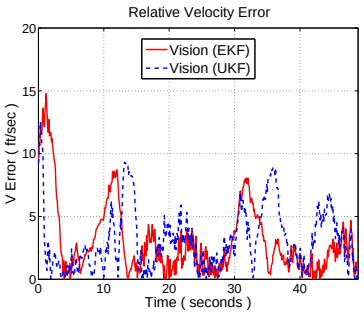


(c) Z Relative Position Error (ft)

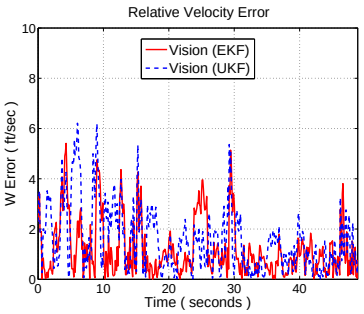
Figure 4.14: Relative Position Error



(a) U Relative Velocity Error (ft/sec)

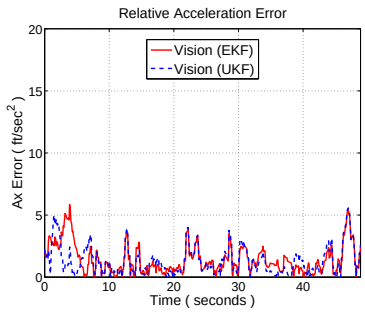


(b) V Relative Velocity Error (ft/sec)

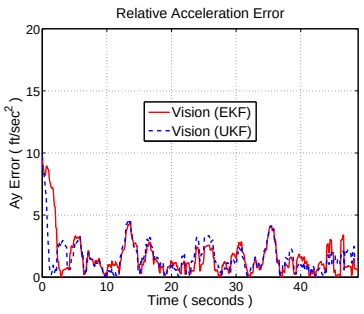


(c) W Relative Velocity Error (ft/sec)

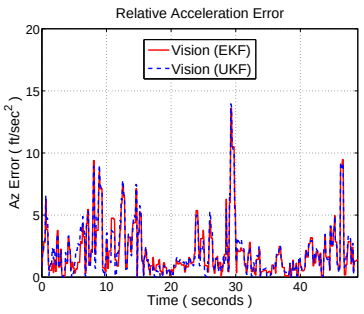
Figure 4.15: Relative Velocity Error



(a) A_x Relative Acceleration Error (ft/sec^2)

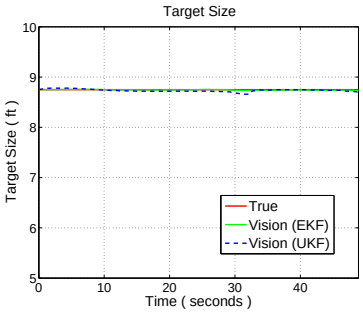


(b) A_y Relative Acceleration Error (ft/sec^2)

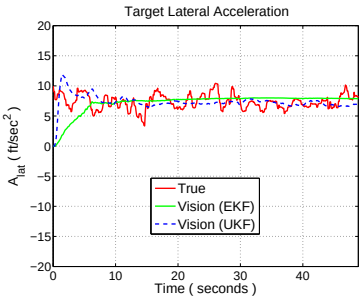


(c) A_z Relative Acceleration Error (ft/sec^2)

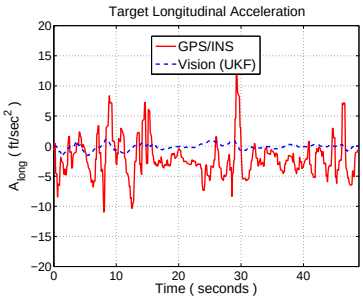
Figure 4.16: Relative Acceleration Error



(a) Target Size, b (ft)



(b) Target Lateral Acceleration, a_{lat} (ft/sec^2)



(c) Target Longitudinal Acceleration, a_{long} (ft/sec^2)

Figure 4.17: Target Characteristics

4.4 Summary

This chapter discusses the development of a vision-based tracking system to be used in the research of autonomous research UAV's and general air vehicles. The vision-based tracking system estimates the relative kinematics and some target characteristics based on vision-only target image information in the framework of an unscented Kalman filter (UKF). The vision information for the relative motion estimator is composed of three target angles: the azimuth angle, the elevation angle, and the subtended angle. Using these measurements of vision information, the ownship aircraft estimates the relative position, the relative velocity, the size, and the lateral/longitudinal acceleration components of the target using the UKF, which is applied to the relative motion estimator due to the highly nonlinear characteristics of the problem at hand. By incorporating some of target characteristics such as the target size and target lateral/longitudinal accelerations as part of the estimation states, accurate estimation of the relative motion kinematics about the target is obtained. Accurate estimation of the surrounding environment (e.g., relative location/velocity of stationary or moving obstacles, relative position/velocity of friendly or enemy vehicles) can provide better probability of ownship safety and a successful mission under the circumstances of noisy and uncertain environment information.

In order to evaluate the performance of the vision-based tracking filter, vision

information obtained through a real-time flight test is post-processed by the filter. The real-time vision information, obtained by a geometric active contour method on an onboard computer, is composed of three points of the target aircraft on the camera image plane. These target points are the center point, the left wingtip point, and the right wingtip point. The target center point on the camera image plane provides information about the azimuth and the elevation angles of the target in the camera frame, and the two wingtip points provide information about the subtended angle of the target that provides the target size. The vision-based estimation results about the target-ownship relative motion and target characteristics are compared to actual data that are independently obtained from onboard integrated navigation systems of both aircraft during the flight test. Comparisons indicate that the vision-based estimation filter provides satisfactory estimation results and thus successfully overcomes the highly nonlinear system characteristics in the UKF framework.

CHAPTER 5

VISION-BASED TRACKING SYSTEM BASED ON EXTENDED MARGINALIZED PARTICLE FILTER

5.1 Introduction

As the target tracking sensor in this work, we use the vision sensor system that is typically the combination of hardware equipment (e.g., a video camera, a camera-image capturing hardware or frame-grabber to capture still images from a video stream, an onboard computer to process the computer vision algorithms) and computer vision software (e.g., captured-image processing software to extract target image features) [42]. The use of a vision sensor as the tracking sensor of low-cost UAV's is very attractive since it is a very efficient sensor due to its compact size and reduced cost while providing rich information. On the other hand, one of the main impediments which make it difficult to use the vision information is the involved computational burden for image-processing in order to extract useful information from the sequence of real-time images obtained from a camera or cameras. Nevertheless, recent progress in both hardware and software technology has made feasible to use the vision system in complicated UAV missions. In terms of hardware and software, first, a frame-grabber captures the sequence of high-rate still images from the image stream of a video camera in real time, and then this sequence of still images is image-processed and feature-extracted in an onboard computer by using recent computer vision algorithms such as geometric active contours [43, 45]. Finally, image processing is generally followed by a vision-based state estimator which extracts specific environment information depending on the sophistication of UAV missions.

The goal of vision-based tracking research in this chapter is to develop a particle

filtering framework for a vision-based tracking system in which probabilistic target information from an image processing algorithm can be incorporated as the vision sensor observations to the vision-based tracking system (or vision-based relative motion estimator).

Even though particle filters have many attractive features including their applicability to general nonlinear, non-Gaussian problems without approximating noise probability distributions, they also suffer from some defects. The most serious defect might be the increasing computational cost in high-dimensional state-space models because a huge number of samples are required in order to appropriately approximate the posterior distributions. One technique to overcome this problem without reducing the efficiency of sampling techniques is to reduce the dimension of the state space model by marginalizing out some of the state variable components. The marginalization (also called Rao Blackwellization) has generally been applied to the state variable components that can be expressed by linear dynamics with Gaussian noise and thus can be handled by the linear Kalman filters [31, 32, 37, 89, 109]. In this work, the idea of marginalizing out some state components can be effectively extended even to the state components that are expressed by nonlinear dynamics. While part of the state components are represented by nonlinear dynamics with Gaussian noise, those state components can be effectively marginalized out by employing the unscented Kalman filter (UKF) to estimate those components. The idea utilizes the reasoning that the UKF can more accurately and effectively solve the nonlinear estimation problems with Gaussian noise characteristics compared to the extended Kalman filter (EKF). Since many real-world problems are composed of high-dimensional state-space models, and, at the same time, generally can only be represented by highly nonlinear dynamics, incorporating the UKF to solve part of the nonlinear dynamics allows us to solve many important real-world problems in the particle filtering framework.

The vision-based tracking problem could be effectively handled in the particle

filtering framework by applying the extended marginalized particle filter (EMPF) described in Section 2.6. Since vision sensor measurements can better be represented by the non-Gaussian noise characteristics, the vision-based tracking problem can be more effectively solved in the particle filtering framework. However, since the problem can only be completely described by a relatively high-dimensional state-space model, direct employment of the particle filtering on this problem is almost impossible because an enormous number of samples are required to properly approximate the posterior distributions. Nevertheless, since the vision information itself directly provides the position information only (and not directly but indirectly the velocity and acceleration information over the progression of time), only the position state components with measurements of vision information are solved in the particle filtering framework, and the other state components represented by nonlinear equations with Gaussian noise are handled by the UKF. This framework can be easily extended to the design of a vision-based tracking system that incorporates probabilistic non-Gaussian vision information.

5.2 System Description for Vision-Based Relative Navigation

This section describes the formulation details of a vision-based relative motion estimator using an extended marginalized particle filter (EMPF). The state vector, composed of the relative position, the relative velocity, the size, and the lateral and longitudinal acceleration components of the target, is decomposed into two parts: one that will be estimated by the particle filtering framework and the other by the unscented Kalman filtering framework. Three target angle-related quantities–the tangent of the azimuth angle, the tangent of the elevation angle, and the tangent of half of the subtended angle–are used as the measurements obtained from the onboard image processor. By using the EMKF, we are able to effectively solve the vision-based tracking problem in the particle filtering framework. In this framework, we can incorporate

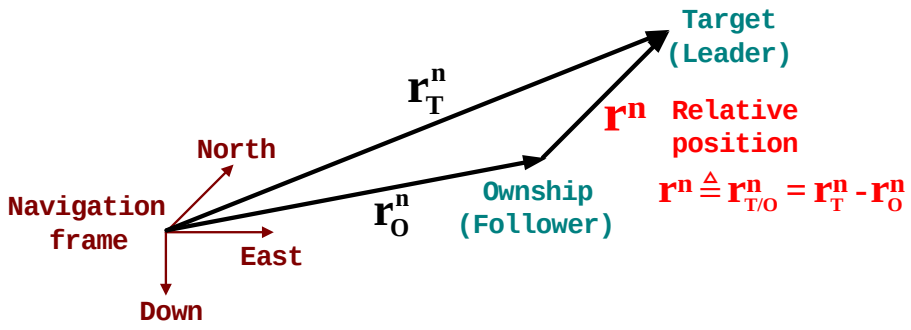


Figure 5.1: Target Relative Position in a Navigation Frame

even the vision information with non-Gaussian measurement characteristics.

5.2.1 General State Space Formulation for Vision-Based Tracking

Let's start with the summary of the general state-space formulation for the vision-based tracking problem described in detail in the previous chapter and then decompose the state components by marginalization or Rao-Blackwellization in order to apply the extended marginalized particle filter (EMPF) to this tracking problem. The state vector of the process model for the vision-based relative navigation includes three relative position components, three relative velocity components, a target size, and target lateral and longitudinal acceleration components:

$$\mathbf{x} = \begin{bmatrix} x_{1:3} \\ x_{4:6} \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} \mathbf{r}^n \\ \mathbf{v}^n \\ b \\ a_{lat} \\ a_{long} \end{bmatrix}, \quad \text{where } \mathbf{r}^n = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \ \mathbf{v}^n = \begin{bmatrix} U \\ V \\ W \end{bmatrix}. \tag{5.1}$$

Here, as represented in Figure 5.1, $\mathbf{r}^n$ and $\mathbf{v}^n$ are a relative position vector and a relative velocity vector from an ownship vehicle to a target vehicle in a navigation frame, respectively, b is the target size, a_{lat} is the target lateral acceleration, and a_{long} is the target longitudinal acceleration.

From Chapter 4, we have the following general process model for a vision-based

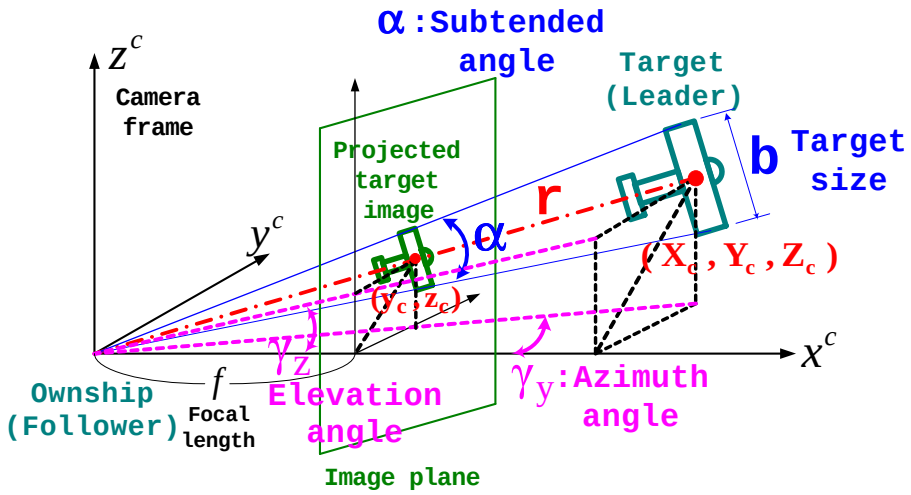


Figure 5.2: Projected Target Image on the Image Plane of a Pinhole Camera

tracking problem in continuous-time state-space matrix form:

$$\dot{\mathbf{x}}(\mathbf{t}) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{w}(t) \tag{5.2}$$

or

$$\begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{b} \\ \dot{a}_{lat} \\ \dot{a}_{long} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ \mathbf{a}^n \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{n}_v \\ \mathbf{n}_a \\ n_b \\ n_{alat} \\ n_{along} \end{bmatrix}, \tag{5.3}$$

where

$$\mathbf{v}^n = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix}, \quad \mathbf{a}^n = \mathbf{a}_T^n - \mathbf{a}_O^n = \frac{1}{\sqrt{U_T^2 + V_T^2}} \begin{bmatrix} \frac{U_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & -V_T \\ \frac{V_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & U_T \\ -\frac{U_T^2+V_T^2}{\sqrt{U_T^2+V_T^2+W_T^2}} & 0 \end{bmatrix} \begin{bmatrix} x_9 \\ x_8 \end{bmatrix} - \mathbf{a}_O^n, \tag{5.4}$$

$$\begin{bmatrix} U_T \\ V_T \\ W_T \end{bmatrix} = \mathbf{v}_T^n = \mathbf{v}^n + \mathbf{v}_O^n = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} + \mathbf{v}_O^n. \tag{5.5}$$

In a matrix component form, we have

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \frac{1}{\sqrt{U_T^2+V_T^2}} \begin{bmatrix} \frac{U_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & -V_T \\ \frac{V_T W_T}{\sqrt{U_T^2+V_T^2+W_T^2}} & U_T \\ -\frac{U_T^2+V_T^2}{\sqrt{U_T^2+V_T^2+W_T^2}} & 0 \end{bmatrix} \begin{bmatrix} x_9 \\ x_8 \end{bmatrix} - \mathbf{a}_O^n \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{n}_v \\ \mathbf{n}_a \\ n_b \\ n_{alat} \\ n_{along} \end{bmatrix}, \tag{5.6}$$

where

$$\begin{bmatrix} U_T \\ V_T \\ W_T \end{bmatrix} = \mathbf{v}_T^n = \mathbf{v}^n + \mathbf{v}_O^n = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} + \mathbf{v}_O^n. \tag{5.7}$$

The ownship velocity $\mathbf{v}_O^n$ and the ownship acceleration $\mathbf{a}_O^n$ in the navigation frame are assumed to be known from the ownship navigation system described in Chapter 3.

5.2.2 Measurement Model of Vision-based Tracking System

For the measurement quantities, we use three target angle-related quantities as follows:

$$\tan \gamma_y = \frac{Y_c}{X_c}, \tag{5.8}$$

$$\tan \gamma_z = \frac{Z_c}{X_c}, \tag{5.9}$$

$$\tan \frac{\alpha}{2} = \frac{b}{2\,r} = \frac{b}{2\,\sqrt{X_c^2 + Y_c^2 + Z_c^2}}, \tag{5.10}$$

where γ_y is the target azimuth angle, γ_z is the target elevation angle , and α is the target subtended angle [45, 68]. The three angles are illustrated in Figure 4.6. In

addition, $\mathbf{r}^c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = C_n^c \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = C_n^c \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ is the target relative position in a camera frame, C_n^c is the transformation matrix from the navigation frame to the camera frame which is given by the ownship navigation system, $r = \sqrt{X_c^2 + Y_c^2 + Z_c^2}$ is the relative distance between the ownship vehicle and the target vehicle. As a result, we can express the three measurements as a function of state variables as follows:

$$\mathbf{y} = \begin{bmatrix} \tan\gamma_y \\ \tan\gamma_z \\ \tan\frac{\alpha}{2} \end{bmatrix} = \begin{bmatrix} \frac{Y_c}{X_c} \\ \frac{Z_c}{X_c} \\ \frac{x_7}{2\sqrt{X_c^2+Y_c^2+Z_c^2}} \end{bmatrix} \tag{5.11}$$

where $\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{r}^c = C_n^c\ \mathbf{r}^n = C_n^c \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, and C_n^c is assumed to be known from the ownship navigation system.

5.2.3 Measurement from the Image Processor

From the image processor, we receive three data points in a camera image plane. These are the center point (y_c, z_c), the left wingtip point (y_l, z_l), and the right wingtip point (y_r, z_r) of a target in the image plane.

Using the characteristics of a pinhole camera, we have the following relationships for the target azimuth angle and target elevation angle:

$$\tan\gamma_y = \frac{y_c}{f} = \frac{Y_c}{X_c}, \tag{5.12}$$

$$\tan\gamma_z = \frac{z_c}{f} = \frac{Z_c}{X_c}, \tag{5.13}$$

where f is the focal length of a pinhole camera, $(y_c,\ z_c)$ is the projected target center position on the camera image plane, and $\mathbf{r}^c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$ is the relative target position

in the camera frame. Since the relative distance from the origin of camera frame to the center point of projected target image on the image plane is $\sqrt{f^2 + y_c^2 + z_c^2}$ and the projected target size on the image plane is $\sqrt{(y_l - y_r)^2 + (z_l - z_r)^2}$, the target subtended angle can be expressed as follows:

$$\tan\frac{\alpha}{2} = \frac{\sqrt{(y_l - y_r)^2 + (z_l - z_r)^2}}{2\sqrt{f^2 + y_c^2 + z_c^2}} = \frac{b}{2\,r} = \frac{b}{2\sqrt{X_c^2 + Y_c^2 + Z_c^2}}. \tag{5.14}$$

Finally, the three measurements from the image processor are obtained as follows:

$$\mathbf{y}_m = \begin{bmatrix} \tan\gamma_y \\ \tan\gamma_z \\ \tan\frac{\alpha}{2} \end{bmatrix} = \begin{bmatrix} \frac{y_c}{f} \\ \frac{z_c}{f} \\ \frac{\sqrt{(\frac{y_l}{f} - \frac{y_r}{f})^2 + (\frac{z_l}{f} - \frac{z_r}{f})^2}}{2\sqrt{1 + \left(\frac{y_c}{f}\right)^2 + \left(\frac{z_c}{f}\right)^2}} \end{bmatrix} \tag{5.15}$$

5.2.4 Marginalization or Rao Blackwellization

In order to apply the extended marginalized particle filter (EMPF), described in Section 2.6, to the vision-based tracking problem, we first need to decompose the state vector components into two parts: one that will be solved by the Kalman filters including the EKF or the UKF and the other that will be estimated in the particle filtering framework. Since vision information is, in general, better represented by non-Gaussian measurements and it provides only position information, the position related process equation with the measurements of vision information is estimated in the particle filtering framework and other state components are estimated by the UKF. In this case, the state vector components are decomposed as follows:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^{pf} \\ \mathbf{x}_k^{kf} \end{bmatrix}, \tag{5.16}$$

where

$$\mathbf{x}^{pf} \triangleq \mathbf{r}^n = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad \mathbf{x}^{kf} \triangleq \begin{bmatrix} \mathbf{v}^n \\ b \\ a_{lat} \\ a_{long} \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \\ b \\ a_{lat} \\ a_{long} \end{bmatrix}. \tag{5.17}$$

$\mathbf{r}^n = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ is the target relative position with respect to the ownship in a navigation frame, $\mathbf{v}^n = \begin{bmatrix} U & V & W \end{bmatrix}^T$ is the target relative velocity, b is the target size, and a_{lat}, a_{long} are the target lateral and longitudinal acceleration components in a camera frame, respectively. By applying the state decomposition defined in Eqs.(5.16) and (5.17) to the process model in the Eq. (5.7) and the observation model in the Eq. (5.6), we obtain the decomposed state-space equations in a continuous-time domain:

$$\dot{\mathbf{x}}^{pf} = \mathbf{A}^{pf}\mathbf{x}^{kf} + \mathbf{w}^{pf}, \tag{5.18}$$

$$\dot{\mathbf{x}}^{kf} = \mathbf{f}(\mathbf{x}^{kf}) + \mathbf{w}^{kf}, \tag{5.19}$$

$$\mathbf{y}^{pf} = \mathbf{h}(\mathbf{x}^{pf}) + \mathbf{v}, \tag{5.20}$$

where

$$\mathbf{A}^{pf} = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 2} \end{bmatrix}, \tag{5.21}$$

$$\mathbf{f}(\mathbf{x}^{kf}) = \begin{bmatrix} \frac{1}{\sqrt{U_L^2+V_L^2}} \begin{bmatrix} \frac{U_L W_L}{V_t} & -V_L \\ \frac{V_L W_L}{V_t} & U_L \\ -\frac{U_L^2+V_L^2}{V_t} & 0 \end{bmatrix} \begin{bmatrix} a_{long} \\ a_{lat} \end{bmatrix} - \mathbf{a}_F^n \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} U_L \\ V_L \\ W_L \end{bmatrix} = \mathbf{v}_L^n = \mathbf{v}^n + \mathbf{v}_F^n = \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \mathbf{v}_F^n, \tag{5.22}$$

$$\mathbf{h}(\mathbf{x}^{pf}) = \begin{bmatrix} \frac{Y_c}{X_c} \\ \frac{Z_c}{X_c} \\ \frac{b}{2\sqrt{X_c^2+Y_c^2+Z_c^2}} \end{bmatrix}. \tag{5.23}$$

The ownship velocity $\mathbf{v}_F^n$ and ownship acceleration $\mathbf{a}_F^n$ in the navigation frame are assumed to be known from the ownship navigation system.

If we discretize this continuous-time state-space form by using the first-order Euler algorithm, we have the following state-space form in a discrete-time domain:

$$\mathbf{x}_{k+1}^{pf} = \mathbf{x}_k^{pf} + \Delta t\ \mathbf{A}^{pf}\mathbf{x}_k^{kf} + \Delta t\ \mathbf{w}_k^{pf}, \tag{5.24}$$

$$\mathbf{x}_{k+1}^{kf} = \mathbf{x}_k^{kf} + \Delta t\ \mathbf{f}(\mathbf{x}_k^{kf}) + \Delta t\ \mathbf{w}_k^{kf}, \tag{5.25}$$

$$\mathbf{y}_k^{pf} = \mathbf{h}(\mathbf{x}_k^{pf}) + \mathbf{v}_k \tag{5.26}$$

or

$$\mathbf{x}_{k+1}^{pf} = \mathbf{x}_k^{pf} + \mathbf{A}_k^{pf}\ \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}\ \mathbf{w}_k^{pf}, \tag{5.27}$$

$$\mathbf{x}_{k+1}^{kf} = \mathbf{f}_d(\mathbf{x}_k^{kf}) + \mathbf{G}_k^{kf}\ \mathbf{w}_k^{kf}, \tag{5.28}$$

$$\mathbf{y}_k^{pf} = \mathbf{h}(\mathbf{x}_k^{pf}) + \mathbf{v}_k, \tag{5.29}$$

where

$$\mathbf{A}_k^{pf} = \Delta t\ \mathbf{A}^{pf} = \begin{bmatrix} \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta t & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G}_k^{pf} = \Delta t, \tag{5.30}$$

$$\begin{aligned} \mathbf{f}_d(\mathbf{x}_k^{kf}) &\cong \mathbf{x}_k^{kf} + \Delta t\ \mathbf{f}(\mathbf{x}_k^{kf}) \\ &= \begin{bmatrix} U \\ V \\ W \\ b \\ a_{lat} \\ a_{long} \end{bmatrix}_k + \Delta t \begin{bmatrix} \frac{1}{\sqrt{U_L^2+V_L^2}} \begin{bmatrix} \frac{U_L W_L}{V_t} & -V_L \\ \frac{V_L W_L}{V_t} & U_L \\ -\frac{U_L^2+V_L^2}{V_t} & 0 \end{bmatrix} \begin{bmatrix} a_{long} \\ a_{lat} \end{bmatrix} - \mathbf{a}_F^n \\ 0 \\ 0 \\ 0 \end{bmatrix}_k, \quad \mathbf{G}_k^{kf} = \Delta t, \end{aligned} \tag{5.31}$$

$$\mathbf{h}(\mathbf{x}_k^{pf}) = \begin{bmatrix} \frac{Y_c}{X_c} \\ \frac{Z_c}{X_c} \\ \frac{b}{2\sqrt{X_c^2+Y_c^2+Z_c^2}} \end{bmatrix}_k, \tag{5.32}$$

$$\begin{bmatrix} U_L \\ V_L \\ W_L \end{bmatrix} = \mathbf{v}_L^n = \mathbf{v}^n + \mathbf{v}_F^n = \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \mathbf{v}_F^n, \quad \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{r}^c = C_n^c\ \mathbf{r}^n = C_n^c \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{5.33}$$

and $\mathbf{v}_F^n$, C_n^c are assumed to be known from the ownship navigation system.

The following matrix form will be also used in order to apply the extended marginalized particle filter:

$$\mathbf{x}_{k+1}^{kf} = \mathbf{f}_d(\mathbf{x}_k^{kf}) + \mathbf{G}_k^{kf}\ \mathbf{w}_k^{kf}, \tag{5.34}$$

$$\mathbf{y}_k^{pf} = \mathbf{h}(\mathbf{x}_k^{pf}) + \mathbf{v}_k, \tag{5.35}$$

$$\mathbf{y}_k^{kf} = \mathbf{A}_k^{pf}\ \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}\ \mathbf{w}_k^{pf}, \text{ where } \mathbf{y}_k^{kf} = \mathbf{x}_{k+1}^{pf} - \mathbf{x}_k^{pf}. \tag{5.36}$$

Finally, the extended marginalized particle filter (EMPF) is applied to the following decomposed state-space model:

Particle filter part:

$$\mathbf{x}_{k+1}^{pf} = \mathbf{x}_k^{pf} + \mathbf{A}_k^{pf}\ \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}\ \mathbf{w}_k^{pf}, \tag{5.37}$$

$$\mathbf{y}_k^{pf} = \mathbf{h}(\mathbf{x}_k^{pf}) + \mathbf{v}_k. \tag{5.38}$$

Kalman filter part:

$$\mathbf{x}_{k+1}^{kf} = \mathbf{f}_d(\mathbf{x}_k^{kf}) + \mathbf{G}_k^{kf}\ \mathbf{w}_k^{kf}, \tag{5.39}$$

$$\mathbf{y}_k^{kf} = \mathbf{A}_k^{pf}\ \mathbf{x}_k^{kf} + \mathbf{G}_k^{pf}\ \mathbf{w}_k^{pf}, \tag{5.40}$$

where $\mathbf{y}_k^{kf} = \mathbf{x}_{k+1}^{pf} - \mathbf{x}_k^{pf}$ is the pseudo-measurement from the particle filter part.

5.3 Simulation and Results

5.3.1 Initialization and Noise Covariance Setting

Filter initialization is performed by using the current measurements.

1) Initialization of the State Vector

$$\begin{bmatrix} x_1(t_0) \\ x_2(t_0) \\ x_3(t_0) \end{bmatrix}_{mean}^{pf} = \begin{bmatrix} X(t_0) \\ Y(t_0) \\ Z(t_0) \end{bmatrix} = C_c^n \begin{bmatrix} X_c(t_0) \\ Y_c(t_0) \\ Z_c(t_0) \end{bmatrix}, \tag{5.41}$$

$$\begin{bmatrix} x_1(t_0) \\ x_2(t_0) \\ x_3(t_0) \\ x_4(t_0) \\ x_5(t_0) \\ x_6(t_0) \end{bmatrix}^{kf} = \begin{bmatrix} U(t_0) \\ V(t_0) \\ W(t_0) \\ b(t_0) \\ a_{lat}(t_0) \\ a_{long}(t_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \hat{b}_0 \\ 0 \\ 0 \end{bmatrix}, \tag{5.42}$$

where

$$X_c = \frac{b}{2\,\tan\frac{\alpha}{2}\sqrt{1+\tan^2\gamma_y+\tan^2\gamma_z}} = \frac{\hat{b}_0}{2\,y_3(t_0)\sqrt{1+y_1(t_0)^2+y_2(t_0)^2}}, \tag{5.43}$$

$$Y_c = X_c\,\tan\gamma_y = X_c\,y_1(t_0), \tag{5.44}$$

$$Z_c = X_c\,\tan\gamma_z = X_c\,y_2(t_0). \tag{5.45}$$

($\hat{b}_0$=8.75 ft: estimate of target wingspan or target size)

2) Initialization of the Error Covariance Matrix

$$P^{pf}(t_0) = diag\{\sigma_X^2, \sigma_Y^2, \sigma_Z^2\}, \tag{5.46}$$

$$P^{kf}(t_0) = diag\{\sigma_U^2, \sigma_V^2, \sigma_W^2, \sigma_b^2, \sigma_{a_{lat}}^2, \sigma_{a_{long}}^2\}, \tag{5.47}$$

where $\sigma_X = \sigma_Y = \sigma_Z = 1\ ft$, $\sigma_U = \sigma_V = \sigma_W = 1\ ft/sec$, $\sigma_b = 0.02 * \hat{b}_0\ ft$, $\sigma_{a_{lat}} = 10\ ft/sec^2$, $\sigma_{a_{long}} = 10\ ft/sec^2$.

3) Process Noise

$$Q_k^{pf} = \Delta t\ Q_c^{pf},\ \ Q_c^{pf} = diag\{{\sigma_{\dot{X}}}^2, {\sigma_{\dot{Y}}}^2, {\sigma_{\dot{Z}}}^2\}, \tag{5.48}$$

$$Q_k^{kf} = \Delta t\ Q_c^{kf},\ \ Q_c^{kf} = diag\{{\sigma_{\dot{U}}}^2, {\sigma_{\dot{V}}}^2, {\sigma_{\dot{W}}}^2, {\sigma_{\dot{b}}}^2, {\sigma_{\dot{a}_{lat}}}^2, {\sigma_{\dot{a}_{long}}}^2\}, \tag{5.49}$$

where $\sigma_{\dot{X}} = \sigma_{\dot{Y}} = \sigma_{\dot{Z}} = 7\ ft/sec$, $\sigma_{\dot{U}} = \sigma_{\dot{V}} = \sigma_{\dot{W}} = 5\ ft/sec^2$, $\sigma_{\dot{b}} = 0.01 * \hat{b}_0\ ft/sec$, $\sigma_{\dot{a}_{lat}} = 2\ ft/sec^3$, $\sigma_{\dot{a}_{long}} = 2\ ft/sec^3$.

4) Measurement Noise

$$R^{pf} = diag\{\sigma_{\tan\gamma_y}^2, \sigma_{\tan\gamma_z}^2, \sigma_{\tan\frac{\alpha}{2}}^2\}, \tag{5.50}$$

$$R^{kf} = \Delta t\ diag\{\sigma_{\Delta X}^2, \sigma_{\Delta Y}^2, \sigma_{\Delta Z}^2\}, \tag{5.51}$$

where $\sigma_{\tan\gamma_y} = \tan 5^\circ$, $\sigma_{\tan\gamma_z} = \tan 4^\circ$, $\sigma_{\tan\frac{\alpha}{2}} = \tan\frac{3^\circ}{2}$, $\sigma_{\Delta X} = \sigma_{\Delta Y} = \sigma_{\Delta Z} = 15\ ft$.

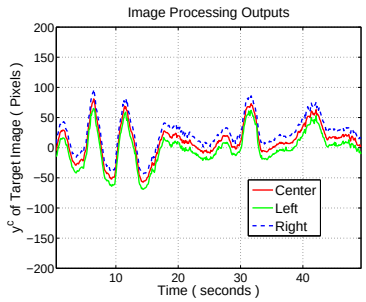
5.3.2 Simulation Results

Similar to the previous chapter, the performance of the vision-based tracking system designed based on the extended marginalized particle filter (EMPF) are tested in the scenario in which an ownship UAV (GTMax) maintains some distance from a target aircraft (GTEdge) based on vision-only information of the target. The navigation of an ownship aircraft relative to a target aircraft is performed by vision-only information from a single camera fixed to the ownship aircraft. The images of the target aircraft projected on the video-camera plane of the ownship aircraft are captured and

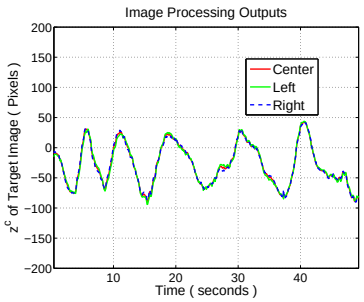
processed into vision information. The estimated states from the image measurements are the target relative position and velocity, the target size, and the target acceleration components. To test the performance of the new vision-based tracking system based on the particle filtering framework, vision information obtained during a real-time flight test is post-processed by the filter. The real-time vision information, obtained by a geometric active contour method on an onboard computer, is composed of three points of the target aircraft on the camera image plane. These target points are the center point, the left wingtip point, and the right wingtip point in the camera image plane presented in Figures 5.3(a) and 5.3(b). Even though current research is to apply to the vision-based tracking system with probabilistic vision information, we use this deterministic vision information since it is the only available realistic vision information at the moment. Once the research on the vision sensor system progresses to provide reliable probabilistic vision information, the work on this chapter can be easily extended to the vision-based tracking system with probabilistic non-Gaussian vision information. Figure 5.5 represents three vision measurements obtained by image processing outputs and used as the inputs to the filter, and Figure 5.4 illustrates their corresponding target angles that are composed of the azimuth angle, the elevation angle, and the subtended angle. Using these vision information measurements, the ownship aircraft estimates the target relative position, the target relative velocity, the target size, and the target acceleration components in the framework of the EMPF described in detail in the Algorithm 2.1. In the actual implementation of the EMPF, 100 particles are used to represent the three position components in the particle filtering part, and the UKF part uses the averaged state vector and error covariance matrix computed by these 100 particles in order to reduce the computational cost instead of independently evolving each UKF described generally in the algorithm.

The vision-based estimation results about the target global motion and target characteristics are compared to actual data that is independently obtained from each

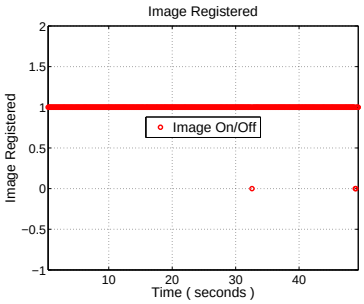
of the aircraft's onboard integrated navigation systems during a flight test. Estimated target global states are compared in Figures 5.6 $\sim$ 5.8. While the target's initial acceleration in the Ref. [45] is assumed to be known ($a_{lat}(t_0) = \frac{V^2}{R}$ by known circular motion) in a formation flight, it is assumed to be unknown and set to zero in this work, which is a more realistic assumption in target tracking problems. In addition, target motion is not confined to horizontal motion (lateral acceleration only) but allowed both horizontal and vertical motion (both lateral and horizontal acceleration) in this work. The target global position in the navigation frame is presented in Figure 5.6. The target position components of the vision-based estimator outputs are compared with the actual flight test results. The actual target position components from the flight test are obtained from the recorded data by the onboard navigation systems (mainly from GPS) of each aircraft. The actual positions of each aircraft are independently recorded during the flight test for comparison purposes. Similarly, the target global velocity components and the acceleration components between the estimator outputs and the flight test results are also compared in the Figures 5.7 and 5.8. The current EMPF-based tracking system provides good performance in estimating the target velocity and acceleration as well as the target position. Since vision information directly provides position information, the error in estimating the target position is not so remarkable in this relatively-inactive horizontal circular flight. Nevertheless, the target velocity and acceleration differ noticeably mainly depending on the capability of capturing the target acceleration. Figure 5.9 compares the target characteristics of the true and the estimated outputs in terms of the target size, the target lateral acceleration, and the target longitudinal acceleration. Figure 5.9(b) shows the rapid convergence in estimating the target lateral acceleration from 0 to around 8 ft/sec^2 without any initial assumption of target motion. Since the flight test was performed in nearly planar circular motion, the target lateral acceleration is nearly constant as



(a) Target on y^c Image Frame (pixels)

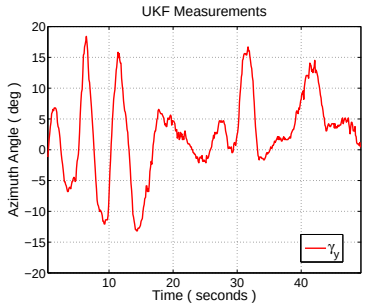


(b) Target on z^c Image Frame (pixels)

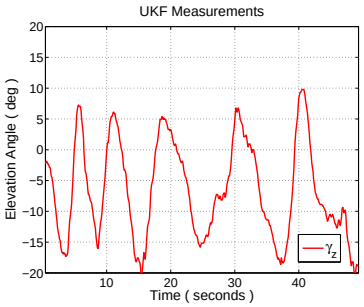


(c) Image Registered

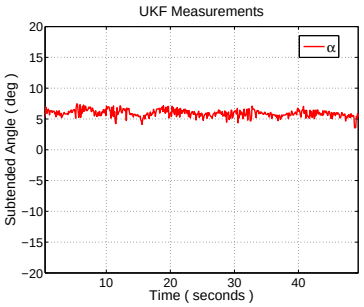
Figure 5.3: Image Processing Outputs of Target Position



(a) Azimuth Angle, γ_y (degrees)



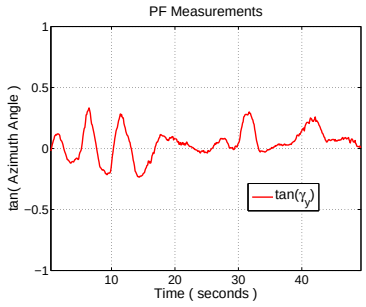
(b) Elevation Angle, γ_z (degrees)



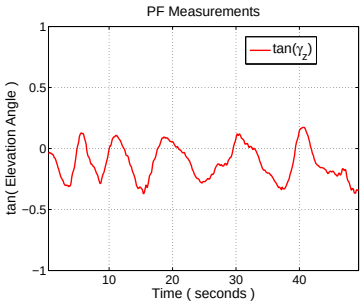
(c) Subtended Angle, α (degrees)

Figure 5.4: Target Angles

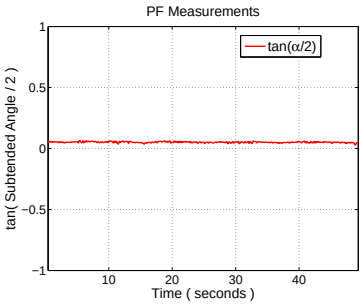
a centripetal acceleration form, and the target longitudinal acceleration is approximately zero. Comparisons indicate that the vision-based estimation filter provides satisfactory estimation results and thus successfully overcomes the highly nonlinear system characteristics by the EMPF framework. Above all, this framework can be easily extended to incorporate vision information with non-Gaussian, multi-modal probability distribution.



(a) tan(Azimuth Angle), $\tan\gamma_y$



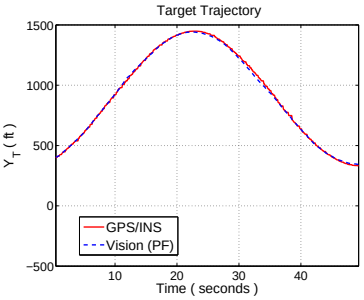
(b) tan(Elevation Angle), $\tan\gamma_z$



(c) tan(Subtended Angle / 2), $\tan\frac{\alpha}{2}$

Figure 5.5: PF Measurements

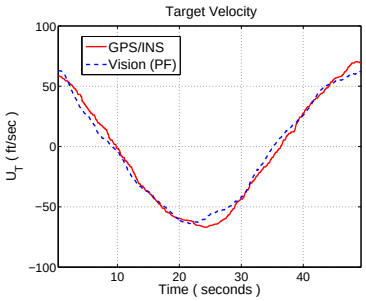
(a) Target Position, X_T



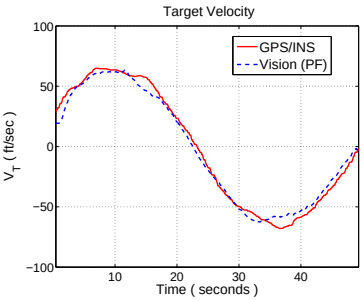
(b) Target Position, Y_T

(c) Target Position, Z_T

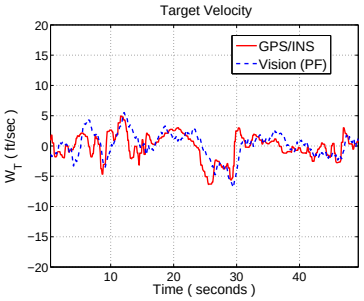
Figure 5.6: Target Trajectory



(a) Target Velocity, U_T

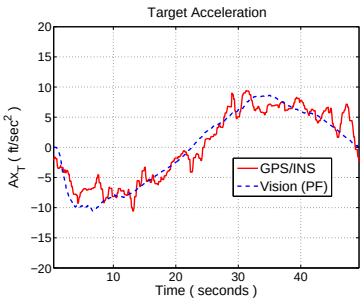


(b) Target Velocity, V_T

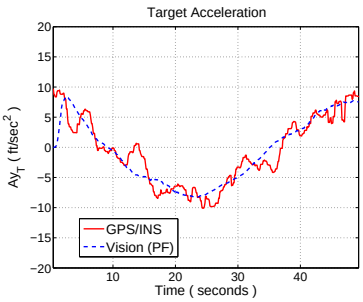


(c) Target Velocity, W_T

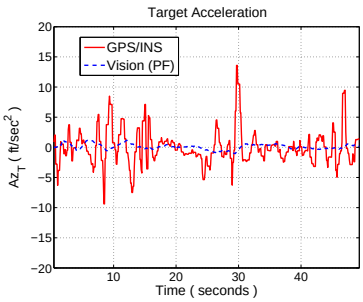
Figure 5.7: Target Velocity



(a) Target Acceleration, Ax_T

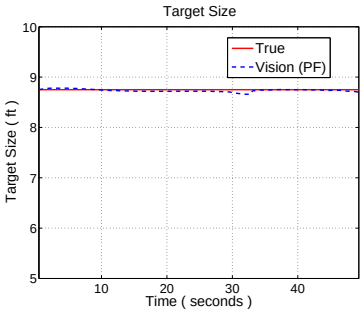


(b) Target Acceleration, Ay_T

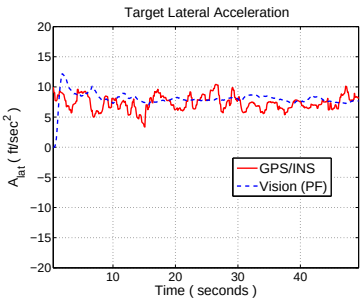


(c) Target Acceleration, Az_T

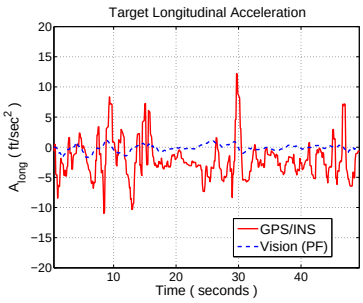
Figure 5.8: Target Acceleration



(a) Target Size, b (ft)



(b) Target Lateral Acceleration, a_{lat} (ft/sec^2)



(c) Target Longitudinal Acceleration, a_{long} (ft/sec^2)

Figure 5.9: Target Characteristics

5.4 Summary

This chapter discusses the new development of a vision-based tracking system based on particle filtering. The vision-based tracking system estimates target relative kinematics and some target characteristics based on vision-only target image information in the framework of the newly-developed extended marginalized particle filter (EMPF). The vision information for the relative motion estimator is composed of three tangent quantities related to the three target angles of the azimuth angle, the elevation angle, and half of the subtended angle. Using this vision information measurement in the EMPF framework, the ownship aircraft estimates the target tracking states including the relative position, the relative velocity, the size, and the lateral/longitudinal acceleration components. By introducing the EMPF, the vision-based tracking system can be designed in the framework of particle filtering. While only three inertial position components are used in the particle filtering part by applying Rao-Blackwellization or marginalization approach, the other six components are used as the state vector components in the unscented Kalman filtering part. Since vision measurement directly provides position information and could be better represented by non-Gaussian noise characteristics, the use of position states with vision information measurement in the particle filtering framework is a natural choice. This research is ultimately aiming to incorporate such non-Gaussian, probabilistic vision information directly as the measurement inputs to the vision-based tracking system. In general, image measurement information can hardly be Gaussian. The nonlinear dynamics included in the other six states (specifically in target acceleration dynamics) could be effectively estimated by introducing the unscented Kalman filter without the necessity of computing the Jacobian matrix.

The vision-based estimation results about the target-ownship relative motion and target characteristics are compared to actual data that is independently obtained from the onboard integrated navigation systems of both aircraft during a flight test.

Comparisons indicate that the new vision-based tracking system provides satisfactory estimation results and thus successfully overcomes the highly nonlinear system characteristics in the EMPF framework. The vision-based tracking system based on the EMPF shows good performance in estimating the target velocity and acceleration as well as the target location.

Even though the estimator performance is simulated in the scenario of the formation flight, this work can be easily applied to not only flying in formation but also avoiding or pursuing other aircraft (e.g., stationary or moving obstacle avoidance, target tracking, evasive maneuvering). As future research work to get better estimation results, we can easily extend this work to incorporate more sophisticated representation of target image information such as the non-Gaussian, probabilistic representation instead of the deterministic three-point representation. In order to improve the particle filtering performance, features of advanced particle filtering algorithms may improve the estimation performance, including the particle regularization, the Markov chain Monte Carlo (MCMC) move step in addition to current sampling importance resampling (SIR) implementation.

CHAPTER 6

CONCLUSIONS AND RECOMMENDED FUTURE RESEARCH

This chapter summarizes the thesis conclusions (or contributions) and the recommended future research.

6.1 Conclusions

1. **Integrated Navigation Systems using the Extended Kalman Filter with Sequential Measurement Updates**

   As a baseline mechanism to estimate the ownship states, an integrated navigation system is designed by using the extended Kalman filter (EKF). In order to effectively fuse various sources of aiding sensor information, the EKF with sequential measurement updates is introduced in the design of the integrated navigation system with a focus on implementation to low-cost autonomous UAV's. Since estimated state accuracy using a low-cost, MEMS-based IMU degrades with time, several absolute (low data rate but absolute error in time) sensors, including the GPS, the magnetometer, and the altimeter, can compensate for time-degrading errors. The original sequential measurement algorithm is developed for the real-time processing of scalar measurements. In this work, the sequential measurement algorithm in small-dimensional measurement-related vectors and matrices is capable of providing a convenient framework for fusing many sources of information in the design of integrated navigation systems. In this framework, several aiding sensor measurements with various sizes and different update rates of information are easily fused into the basic high-rate IMU

processing.

2. **Integrated Navigation Systems using the Unscented Kalman Filter with Sequential Measurement Updates**

   As an advanced mechanism to estimate the ownship states, a new integrated navigation system is designed by using the new unscented Kalman filter that is known to be more accurate and easier to implement. This filter is known to provide at least second-order accuracy by approximating a Gaussian distribution rather than arbitrary nonlinear functions, which is compared to the first-order accuracy of the EKF because of its development based on the first-order term of Taylor series expansion. Moreover, the necessary step of computing the messy Jacobian matrices, always required in the design of an integrated navigation system based on the EKF, is removed. By introducing the new UKF with sequential measurement updates, we can add the advantages of sequential measurement update strategy such as easy handling of sensor latency problems and easy fusion of different size, multi-rate aiding sensor data while maintaining those of the standard UKF such as accurate estimation and removal of Jacobian matrices. Compared to the EKF-based system, the new integrated navigation system based on the UKF with sequential measurement updates is more robust in terms of decreasing estimation error peaks in transient time history and more accurate by reducing steady-state error.

3. **Vision-based Tracking using the Unscented Kalman Filter**

   In order to estimate the target kinematics or target tracking states, a new vision-based tracking system is designed by using the unscented Kalman filter. The tracking system can estimate not only the target tracking states, which includes the target relative position and velocity with respect to an ownship, but also several target characteristics including target size and target lateral/longitudinal

acceleration components. In this framework, the intrinsically highly nonlinear characteristics involved in the vision-based tracking system is successfully handled by the much simpler formulation represented in the Cartesian coordinates without resorting to the complex formulation in modified polar coordinates necessary in the EKF. This is related to the previous simple but important research results in which the UKF generally solves more accurately the generic estimation problem that estimates the position in Cartesian coordinates by using sensor measurements in polar coordinates. Furthermore, by introducing the UKF, the computational step of the messy Jacobian matrices involved in the target acceleration dynamics and angular measurements is removed. The UKF-based tracking system shows better performance in position estimate especially in the initial phase by rapidly estimating target acceleration. If target motion includes vertical acceleration, the difference will become more dramatic since the EKF-based does not include a target vertical acceleration model.

4. **Vision-based Tracking System using the Extended Marginalized Particle Filter**

   A new vision-based tracking system is designed based on particle filtering. The vision-based tracking system estimates target relative kinematics and some target characteristics based on vision-only target image information in the framework of the newly-developed extended marginalized particle filter (EMPF). While only three inertial position components are used in the particle filtering part by applying Rao-Blackwellization or marginalization approach, the other six components are used as state vector components in the unscented Kalman filtering part. Since vision measurements directly provide position information and can be better represented by non-Gaussian noise characteristics, the use of position states with vision measurements in the particle filtering framework is

a natural choice. This research is ultimately aiming to incorporate such non-Gaussian, probabilistic vision information directly as the measurement inputs to the vision-based tracking system. The nonlinear dynamics included in the other states (specifically in target acceleration dynamics) could be effectively handled by introducing the unscented Kalman filter without the necessity of computing the Jacobian matrix. By incorporating some of the target characteristics such as target size and target lateral/longitudinal acceleration components as part of the estimation states, accurate estimation of relative motion kinematics about the target is obtained. The vision-based tracking system based on the EMPF shows good performance in estimating the target velocity and acceleration as well as the target position. Even though the estimator performance is simulated in the scenario of formation flight, this work can be easily applied to not only formation flight but also avoidance or pursuit flight (e.g., stationary or moving obstacle avoidance, target tracking, evasive maneuvering).

5. **Nonlinear Filters (Unscented Kalman Filter with Sequential Measurement Updates, Extended Marginalized Particle Filter)**

   In this research, two new nonlinear filtering methods are suggested, the UKF with sequential measurement updates for developing a new integrated navigation system and the EMPF for developing a new vision-based tracking system. First, the new UKF that combines the original time update step and the sequential measurement update step is developed. This filtering framework provides a convenient tool especially for fusing various sources of different size and multi-rate measurement updates. This filter processes measurements sequentially instead of single processing using big matrices. Therefore, each sensor measurement needs to be treated as just one of several sequential additions depending on its availability at a specific moment. If several sensors are involved in the multi-rate data fusion, determining the dimension of the measurement matrix

depends on the availability of each sensor at the specific time instant. This can involve troublesome coding of the general measurement update equations. The new UKF maintains the advantages of the original UKF including the removal of the messy computational step for the Jacobian matrices as well as better accuracy. Next, a new extended marginalized particle filter (EMPF) is developed. Even though the original marginalization, or Rao-Blackwellization, marginalizes out the linear Gaussian dynamics part, the idea is extended to include the marginalization of nonlinear dynamics with Gaussian noise by incorporating the superior nature of the UKF at solving nonlinear Gaussian estimation problems. This filter is designed to be capable of combining the non-Gaussian measurements with Gaussian but nonlinear substructure dynamics. Parts of the state components that may include non-Gaussian measurement noise are estimated in the particle filtering framework, and the other state components represented by nonlinear dynamics are estimated in the UKF framework.

6.2 Recommended Future Research

1. **Integrated Navigation Systems using Multiplicative Quaternion Approach**

   In terms of the ownship state estimation described in the Chapter 3, the multiplicative quaternion approach may be an alternative choice to improve the estimation performance of the ownship states instead of the additive quaternion approach used in this research.

2. **Vision-based Tracking using Non-Gaussian Probabilistic Vision Information (Extended Marginalized Particle Filter)**

   The extended marginalized particle filter (EMPF) is designed to be capable of combining the non-Gaussian measurements with Gaussian but nonlinear substructure dynamics. Since vision information measurements directly provide

position information and it can be better represented by non-Gaussian noise characteristics, the EMPF is well suited to the vision-based tracking problem. The estimation performance of the target tracking states will be improved by incorporating more sophisticated representations of target image information such as a non-Gaussian, probabilistic representation instead of representing it as just three points.

3. **Nonlinear Filters (Advanced Particle Filters)**

   Particle filtering performance in the EMPF will be improved by incorporating more advanced particle filtering features including the particle regularization, the Markov chain Monte Carlo [3, 12, 23, 100] move step in addition to current sampling importance resampling [23, 33, 100] implementation.

4. **Target motion estimation (Maneuvering Target)**

   In this work, target lateral and longitudinal accelerations are included in the target state estimation with the assumption of constant lateral and longitudinal accelerations. Forward acceleration is set to zero and excluded in the analysis. These assumptions may be a little restrictive to be applied to general target tracking problems. Research on the general target motion analysis is needed to improve the estimation performance of the target tracking states.

APPENDIX A

EKF FORMULATION WITH SEQUENTIAL MEASUREMENT UPDATE

A.1 State Equation and Measurement Update

A.1.1 General Form for State Equation and Measurement Update

The general nonlinear continuous-time process model and discrete-time measurement model in state-space form are given by:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(\mathbf{x}(t), t)\ \mathbf{w}(t), \qquad \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t)), \tag{A.1}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{A.2}$$

where $E[\mathbf{v}_k \mathbf{v}_k^T] = \mathbf{R}_k$.

The measurement update at time $t = t_k^-$ is expressed as follows:

$$\mathbf{K}_k = \mathbf{P}_k^-\ \mathbf{H}_k^T(\hat{\mathbf{x}}_k^-)\left[\mathbf{H}_k(\hat{\mathbf{x}}_k^-)\ \mathbf{P}_k^-\ {\mathbf{H}_k}^T(\hat{\mathbf{x}}_k^-) + \mathbf{R}_k\right]^{-1}, \tag{A.3}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left[\ \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)\ \right], \tag{A.4}$$

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\,\mathbf{P}_k^-, \tag{A.5}$$

where initial conditions are time updated estimates $\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}(t_k^-)$, $\mathbf{P}_k^- = \mathbf{P}(t_k^-)$ and measurement updates $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}(t_k)$, $\mathbf{P}_k = \mathbf{P}(t_k)$ from these relations become the initial conditions for the next time update step.

A.1.2 Sequential Measurement Update Form for State Equation and Measurement Update

When the measurements come from different instruments, the measurement noises at time t_k are usually uncorrelated. In this case, the measurement covariance matrix $\mathbf{R}_k$ becomes block-diagonal, in which each diagonal block corresponds to each

measurement component. Then the nonlinear continuous-time process model and discrete-time measurement model in state-space form are given by:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(\mathbf{x}(t), t)\ \mathbf{w}(t), \qquad \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t)), \tag{A.6}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{A.7}$$

where

$$\mathbf{h}(\mathbf{x}_k, k) = \begin{bmatrix} \mathbf{h}^1(\mathbf{x}_k, k) \\ \cdots \\ \mathbf{h}^r(\mathbf{x}_k, k) \end{bmatrix}, \mathbf{y}_k = \begin{bmatrix} \mathbf{y}_k^1 \\ \cdots \\ \mathbf{y}_k^r \end{bmatrix}, \mathbf{v}_k = \begin{bmatrix} \mathbf{v}_k^1 \\ \cdots \\ \mathbf{v}_k^r \end{bmatrix}, E[\mathbf{v}_k \mathbf{v}_k^T] = \mathbf{R}_k = diag[\mathbf{R}_k^1, \cdots, \mathbf{R}_k^r],$$

and r is the number of aiding sensor measurements. Now, expressing the system equation with the measurement equations in separate form, we get

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(\mathbf{x}(t), t)\ \mathbf{w}(t), \qquad \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t)), \tag{A.8}$$

$$\mathbf{y}_k^l = \mathbf{h}^l(\mathbf{x}_k, k) + \mathbf{v}_k^l, \qquad \mathbf{v}_k^l \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^l), \quad l = 1, 2, \cdots, r, \tag{A.9}$$

where $E[\mathbf{v}_k^i \mathbf{v}_k^{j^T}] = \mathbf{R}_k^i \delta_{ij}$ $(i, j = 1, 2, \cdots, r)$ and δ_{ij} is the Kronecker delta.

Sequential measurement update based on this state space equation is as follows:

For $l = 1, 2, \cdots, r$ (r measurements update at time $t = t_k$),

$$\mathbf{K}_k^l = \mathbf{P}_k^{l-1}\ \mathbf{H}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1}) \left[\mathbf{H}_k^l(\hat{\mathbf{x}}_k^{l-1})\ \mathbf{P}_k^{l-1}\ \mathbf{H}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1}) + \mathbf{R}_k^l\right]^{-1}, \tag{A.10}$$

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l \left[\ \mathbf{y}_k^l - \mathbf{h}^l(\hat{\mathbf{x}}_k^{l-1})\ \right], \tag{A.11}$$

$$\mathbf{P}_k^l = \left[\mathbf{I} - \mathbf{K}_k^l \mathbf{H}_k^l\right] \mathbf{P}_k^{l-1}, \quad l = 1, 2, \cdots, r, \tag{A.12}$$

where initial condition is $\hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}(t_k^-)$, $\mathbf{P}_k^0 = \mathbf{P}_k^- = \mathbf{P}(t_k^-)$ and final measurement update is set as $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}(t_k) = \hat{\mathbf{x}}_k^r$, $\mathbf{P}_k = \mathbf{P}(t_k) = \mathbf{P}_k^r$.

A.2 Derivation of Sequential Measurement Update

Expression for measurement update in covariance matrix ($\mathbf{P}$) are

$$\mathbf{K}_k = \mathbf{P}_k^- \, \mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right]^{-1}, \tag{A.13}$$

$$\mathbf{P}_k = \left[\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right] \mathbf{P}_k^- = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^-, \tag{A.14}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\, \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) \, \right]. \tag{A.15}$$

With slight manipulation we can express the measurement update in information matrix ($\mathbf{P}^{-1}$) as follows [7]:

$$\mathbf{P}_k^{-1} = (\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k, \tag{A.16}$$

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T \mathbf{R}_k^{-1}, \tag{A.17}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\, \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) \, \right]. \tag{A.18}$$

EKF form of sequential measurement update can be derived from this information matrix form.

1) Error Covariance Matrix

$$\mathbf{P}_k^{-1} = (\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \tag{A.19}$$

$$= (\mathbf{P}_k^-)^{-1} + \begin{bmatrix} {\mathbf{H}_k^1}^T & {\mathbf{H}_k^2}^T & \cdots & {\mathbf{H}_k^r}^T \end{bmatrix} \begin{bmatrix} (\mathbf{R}_k^1)^{-1} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & (\mathbf{R}_k^r)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{H}_k^1 \\ \vdots \\ \mathbf{H}_k^r \end{bmatrix} \tag{A.20}$$

$$= (\mathbf{P}_k^-)^{-1} + {\mathbf{H}_k^1}^T (\mathbf{R}_k^1)^{-1} \mathbf{H}_k^1 + {\mathbf{H}_k^2}^T (\mathbf{R}_k^2)^{-1} \mathbf{H}_k^2 + \cdots \tag{A.21}$$

$$= \underbrace{\underbrace{(\mathbf{P}_k^0)^{-1} + {\mathbf{H}_k^1}^T (\mathbf{R}_k^1)^{-1} \mathbf{H}_k^1}_{(\mathbf{P}_k^1)^{-1}} + {\mathbf{H}_k^2}^T (\mathbf{R}_k^2)^{-1} \mathbf{H}_k^2}_{(\mathbf{P}_k^2)^{-1}} + \cdots \tag{A.22}$$

$$\Longleftrightarrow \; (\mathbf{P}_k^l)^{-1} = (\mathbf{P}_k^{l-1})^{-1} + {\mathbf{H}_k^l}^T (\mathbf{R}_k^l)^{-1} \mathbf{H}_k^l, \; l = 1, 2, \cdots, r. \tag{A.23}$$

$$\therefore \; (\mathbf{P}_k^l)^{-1} = (\mathbf{P}_k^{l-1})^{-1} + {\mathbf{H}_k^l}^T (\mathbf{R}_k^l)^{-1} \mathbf{H}_k^l, \; l = 1, 2, \cdots, r. \tag{A.24}$$

2) Kalman Gain

$$\mathbf{K}_k(=\begin{bmatrix}\mathbf{K}_k^1 & \mathbf{K}_k^2 & \cdots & \mathbf{K}_k^r\end{bmatrix}) = \mathbf{P}_k\mathbf{H}_k^T\mathbf{R}_k^{-1} \tag{A.25}$$

$$= \mathbf{P}_k \begin{bmatrix}\mathbf{H}_k^{1^T} & \mathbf{H}_k^{2^T} & \cdots & \mathbf{H}_k^{r^T}\end{bmatrix} \begin{bmatrix}(\mathbf{R}_k^1)^{-1} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & (\mathbf{R}_k^r)^{-1}\end{bmatrix} \tag{A.26}$$

$$= \begin{bmatrix}\mathbf{P}_k\mathbf{H}_k^{1^T}(\mathbf{R}_k^1)^{-1} & \mathbf{P}_k\mathbf{H}_k^{2^T}(\mathbf{R}_k^2)^{-1} & \cdots & \mathbf{P}_k\mathbf{H}_k^{r^T}(\mathbf{R}_k^r)^{-1}\end{bmatrix} \tag{A.27}$$

$$\Longleftrightarrow\ \mathbf{K}_k^l = \mathbf{P}_k^l\mathbf{H}_k^{l^T}(\mathbf{R}_k^l)^{-1},\ l = 1, 2, \cdots, r. \tag{A.28}$$

$$\therefore\ \mathbf{K}_k^l = \mathbf{P}_k^l\mathbf{H}_k^{l^T}(\mathbf{R}_k^l)^{-1},\ l = 1, 2, \cdots, r. \tag{A.29}$$

3) Measurement Update

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left[\ \mathbf{y}_k - \hat{\mathbf{y}}_k^-\ \right] \tag{A.30}$$

$$= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left[\ \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)\ \right] \tag{A.31}$$

$$= \hat{\mathbf{x}}_k^- + \begin{bmatrix}\mathbf{K}_k^1 & \mathbf{K}_k^2 & \cdots & \mathbf{K}_k^r\end{bmatrix} \begin{bmatrix}\mathbf{y}_k^1 - \mathbf{h}^1(\hat{\mathbf{x}}_k^-) \\ \vdots \\ \mathbf{y}_k^r - \mathbf{h}^r(\hat{\mathbf{x}}_k^-)\end{bmatrix} \tag{A.32}$$

$$= \hat{\mathbf{x}}_k^- + \mathbf{K}_k^1\left[\mathbf{y}_k^1 - \mathbf{h}^1(\hat{\mathbf{x}}_k^-)\right] + \mathbf{K}_k^2\left[\mathbf{y}_k^2 - \mathbf{h}^2(\hat{\mathbf{x}}_k^-)\right] + \cdots \tag{A.33}$$

$$= \underbrace{\underbrace{\hat{\mathbf{x}}_k^0 + \mathbf{K}_k^1\left[\mathbf{y}_k^1 - \mathbf{h}^1(\hat{\mathbf{x}}_k^0)\right]}_{\hat{\mathbf{x}}_k^1} + \mathbf{K}_k^2\left[\mathbf{y}_k^2 - \mathbf{h}^2(\hat{\mathbf{x}}_k^0)\right] + \cdots}_{\hat{\mathbf{x}}_k^2} \tag{A.34}$$

$$\Longleftrightarrow\ \hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l\left[\mathbf{y}_k^l - \mathbf{h}^l(\hat{\mathbf{x}}_k^0)\right],\ l = 1, 2, \cdots, r. \tag{A.35}$$

$$\Longrightarrow \hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l\left[\mathbf{y}_k^l - \mathbf{h}^l(\hat{\mathbf{x}}_k^{l-1})\right],\ l = 1, 2, \cdots, r. \tag{A.36}$$

$$\therefore \hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l\left[\mathbf{y}_k^l - \mathbf{h}^l(\hat{\mathbf{x}}_k^{l-1})\right],\ l = 1, 2, \cdots, r. \tag{A.37}$$

We use sequentially updated state ($\hat{\mathbf{x}}_k^{l-1}$) in the measurement estimate equation ($\hat{\mathbf{y}}_k^l = \mathbf{h}^l(\hat{\mathbf{x}}_k^{l-1})$) instead of using the time updated state ($\hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_k^-$) for all sequential measurement update steps. This make sense since we are using more updated states.

Gathering the sequential update equations, we have the following recursive relations.

- Information matrix update form

$$(\mathbf{P}_k^l)^{-1} = (\mathbf{P}_k^{l-1})^{-1} + \mathbf{H}_k^{l\,T}(\mathbf{R}_k^l)^{-1}\mathbf{H}_k^l, \tag{A.38}$$

$$\mathbf{K}_k^l = \mathbf{P}_k^l \mathbf{H}_k^{l\,T}(\mathbf{R}_k^l)^{-1}, \tag{A.39}$$

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l \left[\mathbf{y}_k^l - \mathbf{h}^l(\hat{\mathbf{x}}_k^{l-1})\right],\ l = 1, 2, \cdots, r. \tag{A.40}$$

- Covariance matrix update form

$$\mathbf{K}_k^l = \mathbf{P}_k^{l-1}\ \mathbf{H}_k^{l\,T}\left[\mathbf{H}_k^l \mathbf{P}_k^{l-1} \mathbf{H}_k^{l\,T} + \mathbf{R}_k^l\right]^{-1}, \tag{A.41}$$

$$\mathbf{P}_k^l = \left[\mathbf{I} - \mathbf{K}_k^l \mathbf{H}_k^l\right] \mathbf{P}_k^{l-1}, \tag{A.42}$$

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l \left[\mathbf{y}_k^l - \mathbf{h}^l(\hat{\mathbf{x}}_k^{l-1})\right],\ l = 1, 2, \cdots, r. \tag{A.43}$$

REFERENCES

[1] Aidala, V. J. and Hammel, S. E., "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control*, no. 3, pp. 283–294, 1983.

[2] Anderson, B. D. O. and Moore, J. B., *Optimal Filtering*. Mineola, NY: Dover Publishings, 2005.

[3] Bar-Shalom, Y., Li, X., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation*. New York: John Wiley & Sons, 2001.

[4] Barbour, N., "Inertial components-past, present, and future," AIAA guidance, navigation, and control conference, Montreal, Canada, Aug. 2001. AIAA-2001-4290.

[5] Biezad, D. J., *Integrated Navigation and Guidance Systems*. Reston, VA: AIAA Education Series, 1999.

[6] Brehard, T. and Cadre, J.-P. L., "Distributed target tracking for nonlinear systems: Application to bearings-only tracking," 2005 7th international conference on information fusion FUSION, 2005.

[7] Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*. New York: John Wiley & Sons, 3rd ed., 1997.

[8] Bruno, M. G. S. and Pavlov, A., "Improved particle filters for ballistic target tracking," IEEE international conference on acoustics, speech, and signal processing, 2004. proceedings. (icassp '04), May 2004.

[9] Campbell, M. E. and Wheeler, M., "A vision based geolocation tracking system for uav's," AIAA guidance, navigation, and control conference, Aug. 2006.

[10] Carpenter, J., Clifford, P., and Fearnhead, P., "Improved particle filter for nonlinear problems," *IEE Proceedings-Radar, Sonar Navigation*, vol. 146, pp. 2–7, Feb. 1999.

[11] Carvalho, H., Moral, R. D., Monin, A., and Salut, G., "Optimal nonlinear filtering in gps/ins integration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 835–850, 1997.

[12] Chen, M.-H., Shao, Q.-M., and Ibrahim, J. G., *Monte Carlo Methods in Bayesian Computation*. New York: Springer, 2000.

[13] Chui, C. K. and Chen, G., *Kalman Filtering with Real-Time Applications*. New York: Springer, 3rd ed., 1999.

[14] Crassidis, J. L., "Sigma-point kalman filtering for integrated gps and inertial navigation," AIAA guidance, navigation, and control conference, Aug. 2005.

[15] Crawford, B. G. and Downing, D. R., "Design and evaluation of an autonomous, obstacle avoiding, flight control system using simulated visual sensors," AIAA 3rd unmanned unlimited technical conference, 2004.

[16] Cui, S.-H. and Zhu, C.-Q., "Application of kalman filter to bearing-only target tracking system," international conference on signal processing, ICSP 1996, 1996.

[17] Dale, A. I., *A History of Inverse Probability : From Thomas Bayes to Karl Pearson*. New York: Springer, 1999.

[18] Daum, F., "Nonlinear filters: Beyond the kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, pp. 57–69, Aug. 2005.

[19] Daum, F. and Huang, J., "Curse of dimensionality and particle filters," IEEE aerospace conference, 2003.

[20] Dittrich, J. S., "Design and integration of an unmanned aerial vehicle navigation system," m.s. thesis, georgia institute of technology, May 2002.

[21] Dittrich, J. S. and Johnson, E. N., "Multi-sensor navigation system for an autonomous helicopter," proceedings of the 21st digital avionics systems conference.

[22] Doucet, A., "On sequential simulation-based methods for bayesian filtering," Technical Report CUED/F-INFENG/TR 310, University of Cambridge, 1998.

[23] Doucet, A., de Freitas, N., and Gordon, N., eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[24] Eck, C. and Geering, H. P., "Error dynamics of model based ins/gps navigation for an autonomously flying helicopter," AIAA guidance, navigation, and control conference, Aug. 2000.

[25] El-Sheimy, N., Shin, E.-H., and Niu, X., "Kalman filter face-off: Extended vs. unscented kalman filters for integrated gps and mems inertial," *Inside GNSS*, pp. 48–54, 2006.

[26] Farina, A., Ristic, B., and Benvenuti, D., "Tracking a ballistic target: Comparison of several nonlinear filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 854–867, 2002.

[27] Frew, E. W., *Observer Trajectory Generation for Target-Motion Estimation Using Monocular Vision*. PhD thesis, Stanford University, California, 2003.

[28] Gebre-Egziabher, D., Hayward, R. C., and Powell, J. D., "Design of multi-sensor attitude determination systems," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 627–649, 2004.

[29] Gelb, A., *Applied Optimal Estimation*. Cambridge, MA: The M.I.T. Press, 1974.

[30] Gelfand, A. E. and Smith, F. M., "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990.

[31] Giremus, A., Doucet, A., Calmettes, V., and Tourneret, J.-Y., "A rao-blackwellized particle filter for gps/ins integration," in *2004 IEEE ICASSP*, pp. III 964 – III 967, 2004.

[32] Giremus, A., Tourneret, J.-Y., and Djuric, P. M., "An improved regularized particle filter for gps/ins integration," in *2005 IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, pp. 1013–1017, 2005.

[33] Gordon, N. J., Salmond, D. J., and Smith, A. F. M., "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE Prodeedings F on Radar and Signal Processing*, pp. 107–113, 1993.

[34] Grewal, M. S. and Andrews, A. P., *Kalman Filtering: Theory and Practice Using MATLAB*. New York: John Wiley & Sons, 2nd ed., 2001.

[35] Grossman, W., "Bearings-only tracking: A hybrid coordinate system approach," 30th IEEE conference on decision and control, Brighton, England, 1991.

[36] Gubernatis, J. E., ed., *The Monte Carlo Method in the Physical Sciences: Celebrating the 50th Anniversary of the Metropolis Algorithm*, vol. 690 of *AIP Conference Proceedings (Los Alamos, New Mexico, 9-11 June 2003)*. Melville, NY: American Institute of Physics, june 2003.

[37] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssel, U., Jansson, J., Karlsson, R., and Nordlund, P.-J., "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[38] Harney, H. L., *Bayesian Inference: Parameter Estimation and Decisions*. Berlin, New York: Springer, 2003.

[39] Haykin, S., *Kalman Filtering and Neural Networks*. New York: John Wiley & Sons, 2001.

[40] Hue, C., Cadre, J.-P. L., and Perez, P., "Sequential monte carlo methods for multiple target tracking and data fusion," *IEEE Transactions on Signal Processing*, vol. 50, pp. 309–325, Feb. 2002.

[41] Ito, K. and Xiong, K., "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, pp. 910–927, May 2000.

[42] Ivey, G. F. and Johnson, E. N., "Investigation of methods for target state estimation using vision sensors," AIAA guidance, navigation, and control conference, Aug. 2005.

[43] Johnson, E. N., Calise, A. J., Sattigeri, R., Watanabe, Y., and Madyastha, V., "Approaches to vision-based formation control," 43rd IEEE conference on decision and control, 2004.

[44] Johnson, E. N., Calise, A. J., Tannenbaum, A. R., Soatto, S., Hovakimyan, N., and Yezzi, A. J., "Active-vision control systems for complex adversarial 3-d environments," a tutorial, IEEE american control conference, 2005.

[45] Johnson, E. N., Calise, A. J., Watanabe, Y., Ha, J., and Neidhoefer, J. C., "Real-time vision-based relative navigation," AIAA guidance, navigation, and control conference, Aug. 2006.

[46] Johnson, E. N., Calise, A. J., Watanabe, Y., Ha, J., and Neidhoefer, J. C., "Real-time vision-based relative aircraft navigation," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 4, pp. 707–738, 2007.

[47] Johnson, E. N. and Kannan, S. K., "Adaptive trajectory control for autonomous helicopters," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, 2005.

[48] Johnson, E. N. and Schrage, D. P., "The georgia tech unmanned aerial research vehicle: Gtmax," AIAA guidance, navigation, and control conference, Aug. 2003.

[49] Johnson, E. N. and Schrage, D. P., "System integration and operation of a research unmanned aerial vehicle," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 1, Jan. 2004.

[50] Julier, S. J., "The scaled unscented transformation," proceedings of the american control conference, Anchorage, AK, May 2002.

[51] Julier, S. J., "The spherical simplex unscented transformation," proceedings of the american control conference, Denver, Colorado, 2003.

[52] Julier, S. J. and Uhlmann, J. K., "A consistent, debiased method for converting between polar and cartesian coordinate systems," proceedings of aerosense: the 11th international symposium on aerospace/defense sensing, simulation and controls, Orlando, Florida, Aug. 1997.

[53] Julier, S. J. and Uhlmann, J. K., "A new extension of the kalman filter to nonlinear systems," proceedings of aerosense: the 11th international symposium on aerospace/defense sensing, simulation and controls, Orlando, Florida, Aug. 1997.

[54] Julier, S. J. and Uhlmann, J. K., "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," proceedings of the american control conference, Anchorage, AK, May 2002.

[55] Julier, S. J. and Uhlmann, J. K., "Corrections to "unscented filtering and nonlinear estimation"," *Proceedings of the IEEE*, vol. 92, no. 12, p. 1958, 2004.

[56] Julier, S. J. and Uhlmann, J. K., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[57] Julier, S. J. and Uhlmann, J. K., "Fusion of time delayed measurements with uncertain time delays," proceedings of the american control conference, Portland, OR, 2005.

[58] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H., "A new approach for filtering nonlinear systems," proceedings of the american control conference, Seattle, WA, 1995.

[59] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, no. 3, pp. 477–482, 2000.

[60] Kalman, R. E., "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82.

[61] Karlsson, R. and Gustafsson, F., "Recursive bayesian estimation: Bearings-only applications," *IEE Proceedings-Radar, Sonar Navigation*, vol. 152, pp. 305–313, Oct. 2005.

[62] Kayton, M. and Fried, W. R., *Avionics Navigation Systems*. New York: John Wiley & Sons, 2nd ed., 1997.

[63] King, A. D., "Inertial navigation-forty years of evolution," *GEC Review*, vol. 13, no. 3, pp. 140–149, 1998.

[64] Kingston, D. B., "Implementation issues of real-time trajectory generation on small uavs," 2004.

[65] Kurdila, A., Nechyba, M., Prazenica, R., Dahmen, W., Binev, P., DeVore, R., and Sharpley, R., "Vision-based control of micro-air-vehicles: Progress and problems in estimation," 43rd IEEE conference on decision and control, 2004.

[66] Kwok, N. M., Liu, D. K., Fang, G., and Dissanayake, G., "Path planning for bearing-only simultaneous localisation and mapping," IEEE conference on robotics, automation and mechatronics, 2004.

[67] Langelaan, J. and Rock, S., "Navigation of small uavs operating in forests," AIAA guidance, navigation, and control conference, Aug. 2004.

[68] Langelaan, J. W., *State Estimation for Autonomous Flight in Cluttered Environments*. PhD thesis, Stanford University, California, 2006.

[69] Lee, D.-J. and Alfriend, K. T., "Adaptive sigma point filtering for state and parameter estimation," AIAA guidance, navigation, and control conference, Aug. 2004.

[70] Lefebvre, T., Bruyninckx, H., and Schutter, J. D., "Comment on 'a new method for the nonlinear transformation of means and covariances in filters and estimators'," *IEEE Transactions on Automatic Control*, no. 8, pp. 1406–1408, 2002.

[71] Lerro, D. and Bar-Shalom, Y., "Bias compensation for improved recursive bearings-only target state estimation," american control conference, 1995.

[72] Levesque, J.-F., "Second-order simplex sigma points for nonlinear estimation," AIAA guidance, navigation, and control conference, Aug. 2006.

[73] Lewis, F. L., *Optimal Estimation*. New York: John Wiley & Sons, 1986.

[74] Li, X. R. and Jilkov, V. P., "A survey of maneuvering target tracking-part ii: Ballistic target models," proceedings of SPIE conference on signal and data processing of small targets, San Diego, CA, July-Aug. 2001.

[75] Li, X. R. and Jilkov, V. P., "A survey of maneuvering target tracking-part iii: Measurement models," proceedings of SPIE conference on signal and data processing of small targets, San Diego, CA, July-Aug. 2001.

[76] Li, X. R. and Jilkov, V. P., "A survey of maneuvering target tracking-part iv: Decision-based methods," proceedings of SPIE conference on signal and data processing of small targets, Orlando, FL, 2002.

[77] Li, X. R. and Jilkov, V. P., "Survey of maneuvering target tracking. part i: Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, pp. 1333–1364, Oct. 2003.

[78] Li, X. R. and Jilkov, V. P., "A survey of maneuvering target tracking: Approximation techniques for nonlinear filtering," proceedings of SPIE conference on signal and data processing of small targets, Orlando, FL, 2004.

[79] Li, X. R. and Jilkov, V. P., "Survey of maneuvering target tracking. part v: Multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 1255–1321, Oct. 2005.

[80] Lin, C.-F., *Modern Navigation, Guidance, and Control Processing*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[81] Liu, J. S., *Monte Carlo Strategies in Scientific Computing*. New York: Springer, 2001.

[82] Madyastha, V. K., *Adaptive Estimation for Control of Uncertain Nonlinear Systems with Applications to Target Tracking*. PhD thesis, Georgia Institute of Technology, Georgia, 2005.

[83] Martin, M. K. and Vause, D. A., "New low cost avionics with ins/gps for a variety of vehicles," *IEEE Aerospace and Electronic Systems Magazine*, vol. 13, no. 11, pp. 41–47, 1998.

[84] Maybeck, P. S., *Stochastic Models, Estimation, and Control*, vol. 1–3. New York: Academic Press, 1979.

[85] Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J., "Interacting multiple model methods in target tracking: A survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 103–123, Jan. 1998.

[86] McGee, L. A. and Schmidt, S. F., "Discovery of the kalman filter as a practical tool for aerospace and industry," Technical Memorandum NASA TM-86847, National Aeronautics and Space Administration, November 1985.

[87] McLean, S., Maus, S., Dater, D., Macmillan, S., Lesur, V., and Thomson, A., *The US/UK World Magnetic Model for 2005-2010*, 2004. http://www.ngdc.noaa.gov/seg/WMM/data/TRWMM_2005.pdf.

[88] Minvielle, P., "Decades of improvements in re-entry ballistic vehicle tracking," *IEEE A&E Systems Magazine*, vol. 20, pp. 3–14, Aug. 2005.

[89] Nordlund, P.-J. and Gustafsson, F., "Sequential monte carlo filtering techniques applied to integrated navigation systems," in *Proc. of the American Control Conference*, pp. 4375 – 4380, 2001.

[90] Nørgaard, M., Poulsen, N. K., and Ravn, O., "New developments in state estimation for nonlinear systems," *Automatica*, vol. 36, pp. 1627–1638, 2000.

[91] Nørgaard, M., Poulsen, N. K., and Ravn, O., "Advances in derivative-free state estimation for nonlinear systems," tech. rep. imm-rep-1998-15, revised ed., tech. univ. of denmark, 2004.

[92] Oh, S.-M. and Johnson, E. N., "Development of uav navigation system based on unscented kalman filter," AIAA guidance, navigation, and control conference, Aug. 2006.

[93] Oh, S.-M. and Johnson, E. N., "Relative motion estimation for vision-based formation flight using unscented kalman filter," AIAA guidance, navigation, and control conference, Aug. 2007.

[94] Ohlmeyer, E. J., Pepitone, T. R., and Miller, B. L., "Assesment of integrated gps/ins for the ex-171 extended range guided munition," AIAA guidance, navigation, and control conference, Boston, MA, Aug. 1998. AIAA-98-4416.

[95] Orton, M. and Fitzgerald, W., "A bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Transactions on Signal Processing*, vol. 50, pp. 216–223, Feb. 2002.

[96] Oshman, Y. and Koifman, M., "Robust, imm-based, tightly-coupled ins/gps in the presence of spoofing," AIAA guidance, navigation, and control conference, Aug. 2004.

[97] Peach, N., "Bearings-only tracking using a set of range-parameterised extendedkalman filters," *IEE Proceedings-Control Theory and Applications*, vol. 142, pp. 73–80, Jan. 1995.

[98] Proctor, A. A. and Johnson, E. N., "Vision-only aircraft flight control methods and test results," AIAA guidance, navigation, and control conference, Aug. 2004.

[99] Proctor, A. A. and Johnson, E. N., "Vision-only approach and landing," AIAA guidance, navigation, and control conference, Aug. 2005.

[100] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA: Artech House Publishers, 2004.

[101] Ristic, B., Arulampalam, S., and Gordon, N., "Chapter 6. bearings-only tracking," in *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, pp. 103–151, Boston: Artech House, 2004.

[102] Robert, C. P. and Casella, G., *Monte Carlo Statistical Methods*. New York: Springer, 2nd ed., 2004.

[103] Rogers, R. M., *Applied Mathematics in Integrated Navigation Systems*. Reston, VA: AIAA Education Series, 2nd ed., 2003.

[104] Sattigeri, R., *Adaptive Estimation and Control with Application to Vision-Based Autonomous Formation Flight*. PhD thesis, Georgia Institute of Technology, Georgia, Aug. 2007.

[105] Sattigeri, R., Calise, A. J., Kim, B. S., Volyanskyy, K., and Kim, N., "6-dof nonlinear simulation of vision-based formation flight," AIAA guidance, navigation, and control conference, Aug. 2005.

[106] Schei, T. S., "A finite-difference method for linearization in nonlinear estimation algorithms," *Automatica*, vol. 33, no. 11, pp. 2053–2058, 1997.

[107] Schmidt, G. T., "Gps/ins technology trends for military systems," AIAA guidance, navigation, and control conference, New Orleans, LA, Aug. 1997. AIAA-97-3826.

[108] Schmidt, S. F., "The kalman filter: Its recognition and development for aerospace applications," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, 1981.

[109] Schon, T., Gustafsson, F., and Nordlund, P.-J., "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2297–2289, 2005.

[110] Shaviv, I. G. and Oshman, Y., "Guidance without assuming separation," AIAA guidance, navigation, and control conference, Aug. 2005.

[111] Stengel, R. F., *Optimal Control and Estimation*. New York: Dover Publishings, 1994.

[112] Tanizaki, H., *Nonlinear Filters: Estimation and Applications*. New York: Springer, 2nd ed., 1996.

[113] Tenne, D. and Singh, T., "The higher order unscented filter," proceedings of the american control conference, Denver, Colorado, 2003.

[114] Titterton, D. H. and Weston, J. L., *Strapdown Inertial Navigation Technology*, vol. 207. Reston, VA: AIAA, 2nd ed., 2004.

[115] van der Merwe, R., *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, Oregon Health & Science University, Oregon, 2004.

[116] van der Merwe, R. and Wan, E. A., "Efficient derivative-free kalman filters for online learning," proceedings of the ESANN, Bruges, Belgium, 2001.

[117] van der Merwe, R. and Wan, E. A., "The square-root unscented kalman filter for state- and parameter-estimation," proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP), Salt Lake City, May 2001.

[118] van der Merwe, R. and Wan, E. A., "Sigma-point kalman filters for probabilistic inference in dynamic state-space models," workshop on advances in machine learning, Montreal, Canada, 2003.

[119] van der Merwe, R. and Wan, E. A., "Sigma-point kalman filters for nonlinear estimation and sensor-fusion - application to integrated navigation," AIAA guidance, navigation, and control conference, Providence, Rhode Island, Aug. 2004.

[120] Vasconcelos, J. F., Oliveira, P., and Silvestre, C., "Inertial navigation system aided by gps and selective frequency contents of vector measurements," AIAA guidance, navigation, and control conference, Aug. 2004.

[121] Vermaak, J., Godsill, S. J., and Perez, P., "Monte carlo filtering for multi-target tracking and data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 309–332, Jan. 2005.

[122] Wagter, C. D. and Mulder, J. A., "Toward vision-based uav situation awareness," AIAA guidance, navigation, and control conference, Aug. 2005.

[123] Wan, E. A. and van der Merwe, R., "The unscented kalman filter for nonlinear estimation," proceedings of IEEE symposium 2000 (AS-SPCC), Lake Louise, Alberta, Canada, Oct. 2000.

[124] Wan, E. A. and van der Merwe, R., "Chapter 7. the unscented kalman filter," in *Kalman Filtering and Neural Networks* (Haykin, S., ed.), New York: John Wiley & Sons, 2001.

[125] Watanabe, Y., Calise, A. J., Johnson, E. N., and Evers, J. H., "Minimum-effort guidance for vision-based collision avoidance," AIAA guidance, navigation, and control conference, Aug. 2006.

[126] Watanabe, Y., Johnson, E. N., and Calise, A. J., "Vision-based approach to obstacle avoidance," AIAA guidance, navigation, and control conference, Aug. 2005.

[127] Watanabe, Y., Johnson, E. N., and Calise, A. J., "Vision-based guidance design from sensor trajectory optimization," AIAA guidance, navigation, and control conference, Aug. 2006.

[128] Wendel, J., Maier, A., Metzger, J., and Trommer, G. F., "Comparison of extended and sigma-point kalman filters for tightly coupled gps/ins integration," AIAA guidance, navigation, and control conference, Aug. 2005.

[129] Wenger, L. and Gebre-Egziabher, D., "System concepts and performance analysis of multi-sensor navigation systems for uav applications," AIAA guidance, navigation, and control conference, Aug. 2003.

[130] Winkler, S., Buschmann, M., Kruger, L., Schulz, H.-W., and Vorsmann, P., "State estimation by multi-sensor fusion for autonomous mini and micro aerial vehicles," AIAA guidance, navigation, and control conference, San Francisco, California, Aug. 2005.

[131] Xu, Y. and Liping, L., "Single observer bearings-only tracking with the unscented kalman filter," international conference on communications, circuits and systems, ICCCAS 2004, 2004.

[132] Yoo, C.-S. and Ahn, I.-K., "Low cost gps/ins sensor fusion system for uav navigation," IEEE digital avionics systems conference, Oct. 2003.

[133] Zarchan, P. and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach*, vol. 208. Reston, VA: AIAA, 2nd ed., 2005.

VITA

Seung-Min Oh received his B.S. and M.S. degrees in aerospace engineering from Seoul National University in Korea in 1985 and 1987, respectively. From 1990 to 2002, he worked as a (senior) research engineer at the Korean Agency for Defense Development in Daejeon, Korea, where his work included aerodynamic modeling, dynamic stability analysis, flight simulation, and flight test analysis of various aerospace vehicles. Since August 2002, he has been with the "Flight Mechanics and Controls" group in the School of Aerospace Engineering at Georgia Tech, where he has studied for his Ph.D. degree. His research interests include nonlinear estimation using the unscented Kalman filters and particle filters, integrated navigation systems, vision-based tracking systems, probabilistic visual tracking systems, guidance and navigation systems for UAV's and rockets, neural network-based adaptive flight control systems, and control applications to autonomous aerospace vehicles such as UAV's, fixed-wing aircraft, and helicopters.